Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations 1. Thesis and Dissertation Collection, all items

1967-06

Computer techniques for system synthesis

Breslin, Michael Joseph

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/26944

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

NPS ARCHIVE
1967
BRESLIN, M.

UNITED STATES
NAVAL POSTGRADUATE SCHOOL

THESIS

COMPUTER TECHNIQUES FOR SYSTEM SYNTHESIS

by

Michael J. Breslin

June 1967

[redacted]

THESIS
B8038

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

[redacted]

COMPUTER TECHNIQUES
FOR SYSTEM SYNTHESIS

by

Michael Joseph Breslin
Lieutenant, United States Navy
B.M.E., Villanova University, 1960

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

June 1967

ABSTRACT

A major hindrance to the use of algebraic techniques for the design of linear control systems is the non-linearity of the equations describing the system. The equations for a general class of control systems are shown to be linear in certain groupings of variables contained in, or added to, the systems. A design technique based on the solution of these linear equations is outlined, and a digital computer program which will accomplish the design is presented.

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

Table of Contents

Section Page

1. Introduction 11
2. Algebraic Methods 13
    - Method of Mitrovic' 14
    - Coefficient Matching 15
3. Algebraic Solution of a General Problem 18
    - Steady-State Error Specification 24
    - Bandwidth Specification 25
    - Multiple Specifications 26
4. Limitations of the Method 29
    - Steady-State Error for a Ramp Input 29
    - Multi-path Feedback 30
5. Sample Solutions 32
6. Conclusions 75

APPENDIX Program COMPTECH 77

List of Tables

Table Page

1. Matrix System of Equations for Compensated Feedback Control System. 23

List of Illustrations

Figure Page

1. ξ and ω_n in the s-Plane 13
2. Basic Feedback Control System 18
3. Basic Feedback Control System with Cascade Compensation 20
4. Control System with Two Feedback Paths 30
5. Block Diagram for Example 5.1 32
6. Block Diagram for Example 5.2 41
7. Block Diagram for Example 5.3 52
8. Block Diagram for Example 5.4 63

Table of Symbols and Abbreviations

s	a complex variable ($s = \sigma + j\omega$)
X(s)	a transfer function; a ratio of polynomials in the variable s.
Xn(s)	the numerator of X(s)
Xd(s)	the denominator of X(s)
Xn_k	the coefficient of the kth power of s in Xn(s)
f(x,y,z)	a function of the variables x, y and z
Kt	variable forward gain
T(s)	forward transfer function
K_f	variable feedback gain
H(s)	feedback transfer function
C(s)	filter compensation transfer function
Ess	steady-state error
G_{db}	gain in decibels specified in a bandwidth requirement
ω	frequency specified in a bandwidth requirement
ξ	damping ratio, cosine of an angle in the s-plane
ω_n	undamped natural frequency, a radial distance in the s-plane

1. Introduction

Linear control system design may be generally described as the synthesis of physical components or systems, active or passive, whose dynamic response will satisfy specified criteria. Present design procedure is essentially mathematical, since the response of dynamic systems is conveniently described by a set of differential equations. The problem is then to:

1) formulate differential equations whose solution will describe the desired response,
2) relate coefficients of the equations to the parameters of a physical system,
3) obtain a solution of the differential equations in terms of these parameters.

Production of systems designed in this manner is frequently economically unfeasible, since components required may be expensive to manufacture in limited numbers. A more practical approach is to utilize readily available components, combining them and adjusting their variable parameters to obtain a system with the desired dynamic response.

Analysis of systems thus synthesized may be accomplished in the time domain by direct solution of the system differential equations, but the more usual approach is to perform analysis in the complex frequency domain (s-plane), by means of the Laplace Transformation. System equations thus transformed may be approached with several well-known analysis techniques, notably the Evan's root locus procedure, or in the frequency domain ($s = j\omega$), the Bode, Nyquist or Nichols methods. Application of these techniques to the design procedure is restricted, however, since they are not easily described analytically, they are essentially graphical and therefore time-consuming, and the approach to design is largely trial and error. All suffer the further disadvantage that changes in system response may be predicted on the basis of variation of only one parameter.

Several recent techniques, to be discussed in further detail, have succeeded in overcoming the more obvious difficulties listed above. The transformed differential equations describing the system and the specifications prescribing system response are reduced to a set of algebraic functions of the variable parameters. The algebraic relationships produced, however, are non-linear, and for complex systems may prove difficult or impossible to solve. This paper outlines a numerical procedure for the solution of these non-linear equations, and presents a digital computer program which obtains solutions for a general class of control device problems.

2. Algebraic Methods

Application of algebraic techniques requires the reduction of the transformed system equations to a "transfer function," usually the ratio of two polynomials in the variable s, which relates the output of the system to an arbitrary input. The roots of the denominator polynomial, known as the "characteristic equation," determine the stability and response time of the system, factors of principal concern to the designer. For the system to be stable, i.e. finite response to a finite input, it is necessary that all roots of the characteristic equation have negative real parts. It can be shown that the time response of the system is determined by the values of the roots, and is frequently dominated by the root or pair of roots whose real parts have the least negative value, i.e. when plotted in the s-plane, the roots located closest to the origin. The dominant roots are designated in polar coordinates by their radial distance from the origin (ω_n), and the cosine (ξ) of the angle formed by the second quadrant radius vector and the negative real axis.

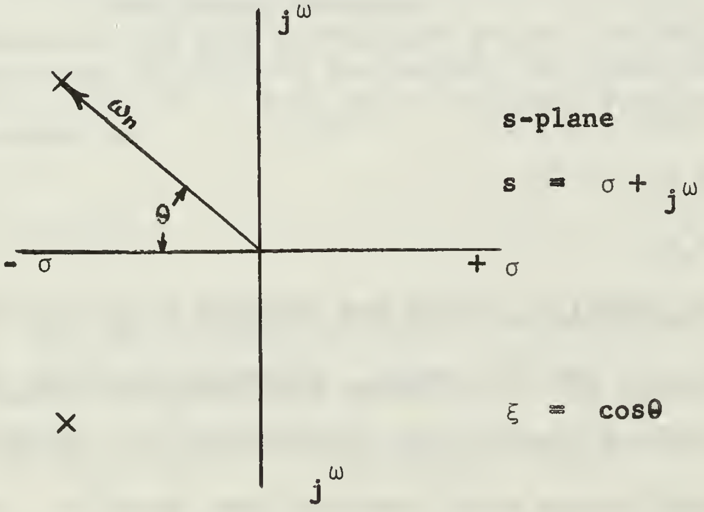

Figure 1

ξ and ω_n in the s-Plane

Method of Mitrovic'

The characteristic equation is a polynomial of the form

$$F(s) = F_n s^n + F_{n-1} s^{n-1} + ---- + F_2 s^2 + F_1 s^1 + F_o \qquad (2.1)$$

or in summation notation

$$F(s) = \sum_{k=0}^{n} F_k s^k \qquad (2.2)$$

where the polynomial coefficients (F_k's) are functions of the system parameters. Mitrovic'[1] has shown, through a transform of variable and s-plane mapping procedure, that two coefficients of F(s) may be stated as functions of the remaining coefficients and of the ξ and ω_n of a pair of specified roots. The result of this procedure is two equations of the form

$$\sum_{k=0}^{n} (-1)^k F_k \omega_n^k \phi_{k-1}(\xi) = 0 \qquad (2.3)$$

$$\sum_{k=0}^{n} (-1)^k F_k \omega_n^k \phi_k(\xi) = 0 \qquad (2.4)$$

where $\phi_{k-1}(\xi)$ and $\phi_k(\xi)$ are recurring Chebishev functions, which for a specified ξ may be precomputed from the relationship

$$\phi_{k+1}(\xi) - 2\xi\phi_k(\xi) + \phi_{k-1}(\xi) = 0 \qquad (2.5)$$

with $\phi_o(\xi) = 0$

and $\phi_1(\xi) = -1$

If numerical values are assigned to all but two of the coefficients (F_k) and to ξ and ω_n, equations (2.3) and (2.4) are reduced to a pair of simultaneous equations in two unknown variables, i.e. the coefficients which have not been specified. Mitrovic' originally chose the coefficients F_o and F_1 to be variables, arriving at the relations

$$F_o = f_1 (F_k, \xi, \omega_n) \tag{2.6}$$

$$F_1 = f_2 (F_k, \xi, \omega_n) \tag{2.7}$$

Later, Sikjak generalized the Mitrovic' method to show that a solution can be obtained for any pair of coefficients [2]. Thus, a relationship between the coefficients of the characteristic equation, which are functions of the system parameters, and the location of a desired pair of roots is established. Designating the variable system parameters as $(a_1, a_2, \text{----}\ a_z)$, we have for any two coefficients

$$F_x = f_3(a_1, a_2, \text{---}\ a_z) = f_4(F_k, \xi, \omega_n) \tag{2.8}$$

$$F_y = f_5(a_1, a_2, \text{---}\ a_z) = f_6(F_k, \xi, \omega_n) \tag{2.9}$$

The process may be repeated to specify additional pairs of roots, thus establishing additional relationships of the form of (2.8) and (2.9). If, however, not all roots of the characteristic equation are specified, the method provides no information on the location of undesignated roots, and does not guarantee either stability or the dominance of a desired pair of roots.

Coefficient Matching

If all of the roots of F(s) are designated, the characteristic equation may be stated as

$$F(s) = \prod_{k=1}^{n} (s + r_k) \tag{2.10}$$

where $(r_1, r_2, \text{---}\ r_n)$ are the roots of F(s). The resultant coefficients of the expanded polynomial are functions of the roots.

$$F(s) = f_n(r_1,r_2,\text{---}r_n)s^n + f_{n-1}(r_1,r_2,\text{---}r_n)s^{n-1} + \text{---} + f_2(r_1,r_2,\text{---}r_n)s^2 + f_1(r_1,r_2,\text{---}r_n)s + f_o(r_1,r_2\text{---}r_n) \tag{2.11}$$

where $f_k(r_1,r_2, --- r_n)$ is the sum of the products of all possible combinations of $(r_1,r_2, ---r_n)$ taken k at a time.

For example:

$$F_k(s) = f_k\,(r_1,r_2, ---r_n)$$

$$= (r_1r_2r_3---r_k) + (r_2r_3r_4---r_{k+1}) + (r_{n-k}r_{n-k+1}---r_n)$$

$$+ (r_1r_3r_4---r_{k+1}) + ---- \qquad (2.12)$$

In summation notation

$$F_k(s) = \sum_{i=1}^{x} \left[\pi_{j=i}^{n}\, k\,(r_j)\right] \; ; \; x = \frac{n!}{k!\,(n-k)!} \qquad (2.13)$$

Since all roots are designated, the F_k's are fixed constants. The result is two expressions for F(s)

$$F(s) = \sum_{i=0}^{n} f_i(a_1,a_2---a_z)s^i \qquad \text{Parameters} \qquad (2.14)$$

$$F(s) = \sum_{i=0}^{n} g_i(r_1,r_2---r_n)s^i \qquad \text{Roots} \qquad (2.15)$$

In order that these expressions be identically equal, coefficients of the same powers of s must be equal.

$$F_ks^k = f_k(a_1,a_2,---a_z)s^k = g_k(r_1,r_2,---r_n)s^k \qquad (2.16)$$

The resultant system is composed of n simultaneous equations in z unknowns. Complete solution of the system of equations requires that the number of unknown quantities (z), the variable parameters, be equal to the number of equations (n), the order of the characteristic equation. If insufficient variables are available, additional parameters must be inserted in the problem in order that this requirement be satisfied.

The principal advantage of designation of all roots is the guarantee of stability and dominance of desired roots. By designating a pair of roots whose values give the desired time response, and locating the remaining roots in such a way as to insure dominance of the original pair and stability of the system, the designer completely specifies the time response of the system. The principal disadvantage is the necessity for addition of arbitrary parameters sometimes required to permit solution of the simultaneous equations.

A disadvantage common to all algebraic methods is the inherent non-linearity of the resultant equations. The expressions usually contain products and/or powers of the variable parameters, and assistance of a digital computer is required for solution of all but the most simple cases.

3. Algebraic Solution of a General Problem

Given a dynamic device whose transformed time domain equation may be expressed as the product of a variable gain (Kt) and a transfer function T(s), and a feedback device expressible as a variable feedback gain (Kf) and the transfer function H(s), the transfer function F(s) of the "closed loop" control system comprising the two components, as arranged in Figure 2, is

$$\frac{\text{Output }(s)}{\text{Input }(s)} = F(s) = \frac{Kt\ T(s)}{1 + Kt\ T(s)\ Kf\ H(s)} \tag{3.1}$$

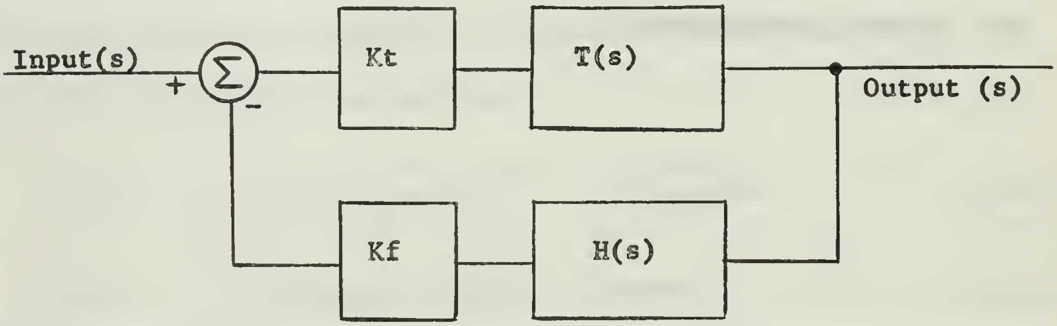

Figure 2

Basic Feedback Control System

Both T(s) and H(s) are the ratio of polynomials in S.

$$T(s) = \frac{Tn(s)}{Td(s)} = \frac{Tn_a s^a + Tn_{a-1} s^{a-1} + \text{----} + Tn_2 s^2 + Tn_1 s + Tn_o}{Td_b s^b + Td_{b-1} s^{b-1} + \text{----} + Td_2 s^2 + Td_1 s + Td_o}$$

$$H(s) = \frac{Hn(s)}{Hd(s)} = \frac{s^c\ (Hn_2 s^2 + Hn_1 s + Hn_o)}{Hd_2 s^2 + Hd_1 s + Hd_o} \tag{3.2}$$

$$\text{and } F(s) = \frac{\dfrac{Kt\ Tn(s)}{Td(s)}}{1 + \dfrac{Kt\ Tn(s)\ Kf\ Hn(s)}{Td(s)\ Hd(s)}} \tag{3.3}$$

F(s) reduces to the ratio of two polynomials

$$F(s) = \frac{Fn(s)}{Fd(s)} = \frac{Kt\ Tn(s)\cdot Hd(s)}{Td(s)\cdot Hd(s) + Kt\ Kf\ Tn(s)\cdot Hn(s)} \qquad (3.4)$$

Forming the product of polynomials where indicated defines:

$$Tn\ Hd(s) = Tn(s)\cdot Hd(s)$$

$$Td\ Hd(s) = Td(s)\cdot Hd(s)$$

$$Tn\ Hn(s) = Tn(s)\cdot Hn(s)$$

The characteristic equation of the system - Fd(s) is the sum of the polynomials TdHd(s) and KtKfTnHn(s).

$$\begin{aligned} Fd(s) &= TdHd(s) + KtKfTnHn(s) \\ &= TdHd_x s^x + TdHd_{x-1} s^{x-1} + \text{----} + TdHd_1 s + TdHd_o \qquad (3.5) \\ &\quad + KtKf(TnHn_y s^y + TnHn_{y-1} s^{y-1} + \text{----} + TnHn_1 s + TnHn_o \end{aligned}$$

In most devices used for control systems, particularly higher order systems, the order of the numerator polynomial is much less than the order of the denominator. Feedback devices seldom have numerator order more than two greater than denominator order, i.e. second derivative feedback. Thus the order of the characteristic equation, and the number of roots to be designated, is determined by the order of the TdHd(s) product. By expanding the designated roots into a polynomial R(s), and equating coefficients, the system of simultaneous equations for Fd(s) becomes

$$\begin{aligned} (TdHd_x)\ s^x &= R_x\ s^x \\ (TdHd_{x-1})\ s^{x-1} &= R_{x-1} s^{x-1} \\ &\vdots \\ (TdHd_y + KtKfTnHn_y)s^y &= R_y s^y \\ (TdHd_{y-1} + KtKfTnHn_{y-1})s^{y-1} &= R_{y-1} s^{y-1} \\ &\vdots \\ (TdHd_2 + KtKfTnHn_2)s^2 &= R_2 s^2 \end{aligned}$$

$$(TdHd_1 = KtKfTnHn_1)s = R_1 s$$

$$(TdHd_o = KtKfTnHn_o) = R_o \qquad (3.6)$$

If all polynomials are rationalized,

$$TdHd_x = R_x = 1$$

and the system consists of x equations in the two unknown parameters Kf and Kt. In order to obtain an algebraic solution, an additional (x-2) variables must be inserted in the system. The addition of "cascade compensation" in the form of a multi-section filter network permits introduction of the necessary mathematical variables and is also an economical method of inserting variable parameters in the physical plant.

Cascade Compensation

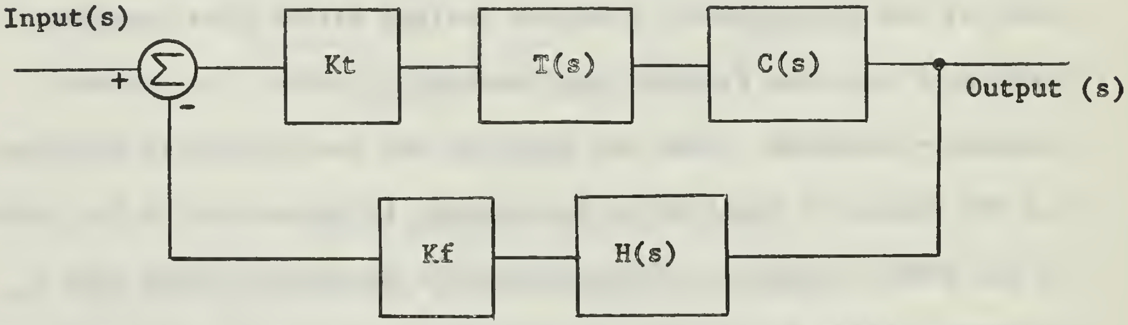

Figure 3

Basic Feedback Control System with Cascade Compensation

The transformed output/input equations C(s) of a resistance-capacitance filter network are expressible as the ratio of two equal order polynomials, the order being equal to the number of filter sections employed.

$$C(s) = \frac{Cn(s)}{Cd(s)} = \frac{Cn_m s^m + Cn_{m-1} s^{m-1} + \text{---} + Cn_2 s^2 + Cn_1 s + Cn_o}{Cd_m s^m + Cd_{m-1} s^{m-1} + \text{---} + Cd_2 s^2 + Cd_1 s + Cd_o} \qquad (3.7)$$

The transfer function of the compensated closed loop system, for the configuration shown in Figure 3, becomes

$$\frac{\text{Output (s)}}{\text{Input (s)}} = F(s) = \frac{Kt\ T(s)\cdot C(s)}{1 + Kt\ T(s)\cdot C(s)\ Kf\ H(s)} \qquad (3.8)$$

which reduces to

$$F(s) = \frac{Fn(s)}{Fd(s)} = \frac{Kt\ TnHd(s)\cdot Cn(s)}{TdHd(s)\cdot Cd(s) + KtKf\ TnHn(s)\cdot Cn(s)} \qquad (3.9)$$

The order of the characteristic equation is now determined by the TdHd(s)•Cd(s) product, thus the addition of each filter stage increases by one; the order of Fd(s), the number of simultaneous equations to be solved, and the number of independent variables required. If the order of TdHd(s) is (x), and (m) filter sections are used, the order of Fd(s) will be (x + m). The addition of each filter section (i) introduces two independent variables into the system, Cn_i and Cd_i. Thus (m) filter sections introduce (2m) variables.

For solution of the system of equations, the product of the independent parameters (KtKf) may be considered as a single algebraic variable. Addition of (m) filter sections to a system whose TdHd(s) is of order (x), produces (x + m) equations in (2m + 1) variables. Equating the number of variables and number of equations

$$x + m = 2m + 1$$

$$m = x - 1 \qquad (3.10)$$

For the system given in Figure 3, the number of filter sections required is determined by the product of the denominator polynomials of the forward and feedback path transfer functions.

The system of simultaneous equations for the compensated system is obtained by designating the required (x + m) roots, expanding the roots to form the polynomial R(s), forming the TdHd(s)•Cd(s) and KtKfTnHn(s)•Cn(s) products, and equating coefficients. Table 1 illustrates, in matrix form, the equations for a system having a third order Td(s)•Hd(s) product, a first order Tn(s)•Hn(s) product, and the required second order filter. The format of the matrix equations is the same for higher order systems if the requirements that

1) The number of filter sections used be one less than the order of Td(s)•Hd(s),

and 2) The order of Tn(s)•Hn(s) be less than the order of Td(s)•Hd(s)

are fulfilled.

Note that the equations for systems of this type are linear in certain individual parameters (Cd_o, Cd_1 --- Cd_m), and in groupings of the other parameters ($KtKfCn_o$, $KtKfCn_1$, --- $KtKfCn_m$). Thus the system of equations may be solved by linear methods for these individual and grouped parameters.

Again, if all polynomials are considered rationalized,

$$TdHd_x \; Cd_m = R_{x+m} = 1$$

so $Cd_m = 1$. If Cn_m is defined as equal to 1, the group variable $KtKfCn_m$ becomes simply the product KtKf. Having solved the linear system and obtained the coefficients of the denominator polynomial (Cd_o, Cd_1, --- Cd_{m-1}) of C(s) directly, the coefficients of the numerator polynomial may be obtained by dividing the grouped variables ($KtKfCn_o$, $KtKfCn_1$, ---- $KtKfCn_{m-1}$), by the value of $KtKfCn_m$, i.e.

Table 1

$$\left\Vert \begin{matrix} TdHd_3 & & & & & \\ TdHd_2 & TdHd_3 & & & & \\ TdHd_1 & TdHd_2 & TdHd_3 & TnHn_1 & & \\ TdHd_o & TdHd_1 & TdHd_2 & TnHn_o & TnHn_1 & \\ & TdHd_o & TdHd_1 & & TnHn_o & TnHn_1 \\ & & TdHd_o & & & TnHn_o \end{matrix} \right\Vert \times \left\Vert \begin{matrix} Cd_2 \\ Cd_1 \\ Cd_o \\ KtKfCn_2 \\ KtKfCn_1 \\ KtKfCn_o \end{matrix} \right\Vert = \left\Vert \begin{matrix} R_5 \\ R_4 \\ R_3 \\ R_2 \\ R_1 \\ R_o \end{matrix} \right\Vert$$

Matrix System of Equations

for

Compensated Feedback Control System

by KtKf. With the numerator and denominator polynomials available, the filter design becomes a problem in passive network synthesis.

A solution has thus been obtained for all physical parameters except the individual gains, Kt and Kf. An additional specification is required in order to obtain an independent solution for values of the gains. The additional criteria considered are steady state error and bandwidth.

Steady - State Error (Ess)

The error of a system is defined as the difference between an input command signal and the system response.

$$\text{Error }(s) = \text{Input }(s) - \text{Output }(s) \tag{3.11}$$

From Equation 3.8

$$\text{Output }(s) = \frac{Kt\ T(s)\cdot C(s)}{1 + Kt\ T(s)\cdot C(s)\ Kf\ H(s)}\ \text{Input }(s)$$

The error as a function of the input signal becomes

$$\text{Error }(s) = \frac{1 + Kt\ T(s)\cdot C(s)[Kf\ H(s) - 1]}{1 + Kt\ T(s)\cdot C(s)\ Kf\ H(s)}\ \text{Input }(s) \tag{3.12}$$

which reduces to

$$\frac{\text{Error }(s)}{\text{Input }(s)} = \frac{TdHd(s)\cdot Cd(s) + KtKf\ TnHn(s)\cdot Cn(s) - KtTnHd(s)\cdot Cn(s)}{TdHd(s)\cdot Cd(s) + KtKf\ TnHn(s)\cdot Cn(s)} \tag{3.13}$$

For a unit step displacement input, Input (s) is 1/s. Error (s) for a step input becomes

$$\text{Error }(s) = \frac{1}{s}\left[1 - \frac{Kt\ TnHd(s)\cdot Cn(s)}{TdHd(s)\cdot Cd(s) + KtKf\ TnHn(s)\cdot Cn(s)}\right] \tag{3.14}$$

Steady-state error is defined as the error of the system after all transient effects have taken place, or mathematically, as time expired after application of the input signal approaches infinity. Application of the Final Value Theorem to Equation 3.14 gives

$$\mathrm{Ess} = \lim_{s \to o} s\,\mathrm{Error}(s)$$

$$\mathrm{Ess} = \lim_{s \to o} \left[1 - \frac{\mathrm{Kt}\ \mathrm{TnHd}(s) \cdot \mathrm{Cn}(s)}{\mathrm{TdHd}(s) \cdot \mathrm{Cd}(s) + \mathrm{KtKf}\ \mathrm{TnHn}(s) \cdot \mathrm{Cn}(s)} \right]$$

$$\mathrm{Ess} = \frac{\mathrm{TdHd}_o \cdot \mathrm{Cd}_o + \mathrm{TnHn}_o\ \mathrm{KtKfCn}_o - \mathrm{TnHd}_o \cdot \mathrm{KtCn}_o}{\mathrm{TdHd}_o \cdot \mathrm{Cd}_o + \mathrm{TnHn}_o \cdot \mathrm{KtKfCn}_o} \qquad (3.15)$$

Note that steady-state error as expressed above is defined for unity input. Error is relative to input, and maximum steady-state error desired in the system is specified as a fraction of the input signal.

Steady-state error is found to be a function of two variables previously defined, Cd_o and KtKfCn_o, and an additional variable, KtCn_o. Values for Cd_o, KtKfCn_o and KtKf having been previously obtained, the solution for $\mathrm{Kt\ Cn}_o$ is

$$\mathrm{Kt\ Cn}_o = \frac{(1 - \mathrm{Ess})\ [\mathrm{TdHd}_o \cdot \mathrm{Cd}_o + \mathrm{TnHn}_o \cdot \mathrm{KtKfCn}_o]}{\mathrm{TnHd}_o} \qquad (3.16)$$

$$\text{and} \quad \mathrm{Kf} = \frac{\mathrm{KtKfCn}_o}{\mathrm{KtCn}_o} \quad ; \quad \mathrm{Kt} = \frac{\mathrm{KtKf}}{\mathrm{Kf}} \qquad (3.17)$$

Values of all unknown parameters have been obtained, and a system is synthesized which will meet specified requirements for stability, root dominance, and steady-state error.

Bandwidth (BW)

The bandwidth of a system is defined as the frequency range over which the response of a control system is within acceptable limits [1]. It is frequently given as that frequency (ω) at which the magnitudes of the output response and the input signal have a specified ratio (G), usually expressed in decibels (db).

If F(s) is the closed loop transfer function of a control system, bandwidth may be found from

$$G_{db} = 20 \log_{10}\left(\frac{\text{output } (_j\omega)}{\text{input } (_j\omega)}\right) = 20 \log_{10}[F(_j\omega)] \qquad (3.18)$$

where ω is the desired bandwidth.

$$G = F(_j\omega) = \frac{Fn(_j\omega)}{Fd(_j\omega)} = \frac{Fn_a(_j\omega)^a + --- + Fn_2(_j\omega)^2 + Fn_1(_j\omega) + Fn_o}{Fd_b(_j\omega)^b + --- + Fd_2(_j\omega)^2 + Fd_1(_j\omega) + Fd_o}$$

$$G = \frac{Fn(\omega)\text{real} + j\, Fn(\omega)\text{imag.}}{Fd(\omega)\text{real} + j\, Fd(\omega)\text{imag.}}$$

$$G = \left[\frac{(Fn(\omega)\text{real})^2 + (Fn(\omega)\text{imag.})^2}{(Fd(\omega)\text{real})^2 + (Fd(\omega)\text{imag.})^2}\right]^{1/2} \qquad (3.19)$$

The bandwidth expression is seen to be highly non-linear, however, this obstacle is not insurmountable. Recalling that

$$F(_j\omega) = \frac{Kt\ TnHd(_j\omega)\cdot Cn(_j\omega)}{TdHd(_j\omega)\cdot Cd(_j\omega) + TnHn(_j\omega)\cdot KtKfCn(_j\omega)}$$

and that values for the variables Cn_i, Cd_i, and $KtKfCn_i$ are available from the solution of the simultaneous equations, Kt may be obtained from the bandwidth criterion. If gain (G) is specified at frequency (ω),

$$Kt = G\left[\frac{TdHd(_j\omega)\cdot Cd(_j\omega) + TnHn(_j\omega)\cdot KtKfCn(_j\omega)}{TnHd(_j\omega)\cdot Cn(_j\omega)}\right] \qquad (3.20)$$

$$\text{and} \qquad Kf = \frac{KtKf}{Kt} \qquad (3.21)$$

Again, values of all unknown parameters are obtained, and the system design will meet stability, root dominance and bandwidth specifications.

Multiple Specifications

Either a steady-state error or a bandwidth specification is sufficient in itself to permit solution for all variable parameters and design of the system. However, if both error and bandwidth are specified, solution for one does not guarantee satisfaction of the other requirement. A slightly different approach must be taken.

No requirement has been placed on the location of the non-dominant, or non-specified roots, other than that they have negative real parts, and that they be sufficiently removed from the origin of the s-plane to insure dominance of the desired roots. This gives the designer a great deal of latitude in the choice of extraneous roots which are added to the problem as a result of cascade compensation.

The system of linear equations used in the solution for parameters is obtained by equating coefficients of the characteristic equation. On one side of each equation is an algebraic function of the variable parameters; on the other side is the numerical coefficient found by expanding the roots to form the polynomial R(s). Changing the location of one or more roots changes the values of the variable parameters. Parameter values obtained are those which will place the roots of the characteristic equation exactly where specified. The steady-state error specification is satisfied by determining the relationship of the forward gain (Kt) to other system parameters. The bandwidth is found from the equation

$$G = \frac{Kt\ TnHd(_j\omega)\ Cn(_j\omega)}{R(_j\omega)} \qquad (3.22)$$

Thus the bandwidth of the system is a function of the location of the roots, but the steady-state error solution is not. By judicious choice of his root locations, the designer may be able to meet both error and bandwidth specifications, or to determine whether a bandwidth specification conflicts with a stability or root dominance criterion.

The process of finding the proper root locations is iterative, laborious, and time-consuming, requiring a complete solution of the system equations after each movement of the roots. For higher order systems, the assistance of a digital computer may be necessary.

4. Limitations of the Method

Steady-State Error for a Ramp Input

Steady-state error, as defined in Section 3, is the limit as s approaches zero of the product of s and the error function Error (s), where

$$\mathrm{Error}(s) = \mathrm{Input}(s)\left[\frac{TdHd(s)\cdot Cd(s) + KtKfTnHn(s)\cdot Cn(s) - KtTnHd(s)\cdot Cn(s)}{TdHd(s)\cdot Cd(s) + KtKf\ TnHn(s)Cn(s)}\right]$$

For a unit ramp input, i.e. constant velocity signal of unity magnitude, Input (s) $= 1/s^2$. Steady-state error becomes

$$Ess = \lim_{s \to 0} \frac{s}{s^2}\left[\frac{TdHd(s)\cdot Cd(s) + KtKfTnHn(s)\cdot Cn(s) - KtTnHd(s)\cdot Cn(s)}{TdHd(s)\cdot Cd(s) + KtKf\ TnHn(s)Cn(s)}\right] \tag{4.1}$$

This limit approaches infinity as s approaches zero, unless an s can be factored from the numerator polynomial of the term in brackets, i.e.

$$\therefore\ TdHd_o \cdot Cd_o + TnHn_o \cdot KtKfCn_o - TnHd_o \cdot KtCn_o = 0 \tag{4.2}$$

Equation 4.2 may be set equal to zero in several ways:

1) The possibility of requiring that Cn_o or Cd_o be zero must be disregarded, as this would fix the value of a variable required for solution of the system of simultaneous equations.

2) Hn_o may be required to be zero, i.e. tachometer feedback, but the steady-state error expression resulting from this procedure introduces two new variables $KtCn_o$ and $KtCn_1$ in a single equation

$$Ess = f(Cd_o,\ Cd_1,\ KtKxCn_o,\ KtCn_o,\ KtCn_1)$$

and solution for independent Kt is impossible.

$$Kt = \frac{Kt\ Cn_o}{Cn_o} \neq Kt = \frac{Kt\ Cn_1}{Cn_1}$$

3) Td_o may be required to be zero, i.e. a pole at the origin of the open loop transfer function. Then

$$Tn_o Hn_o \, KtKfCn_o \; - \; Tn_o Hd_o \, KtCn_o \; = \; 0$$

and $Kf = \frac{Hd_o}{Hn_o}$; i.e. unity feedback.

If this procedure is used Kf is no longer an independent variable and solution is impossible.

4) If Hd_o is required to be zero,

$$Tn_o Hn_o \, KtKfCn_o \; = \; 0$$

or Tn_o must also be zero, and an s factors from the denominator as well as from the numerator of Error (s).

To date, the author has found no expression for steady-state error to a ramp input which is compatible with the system of simultaneous equations previously derived.

Multi-path Feedback

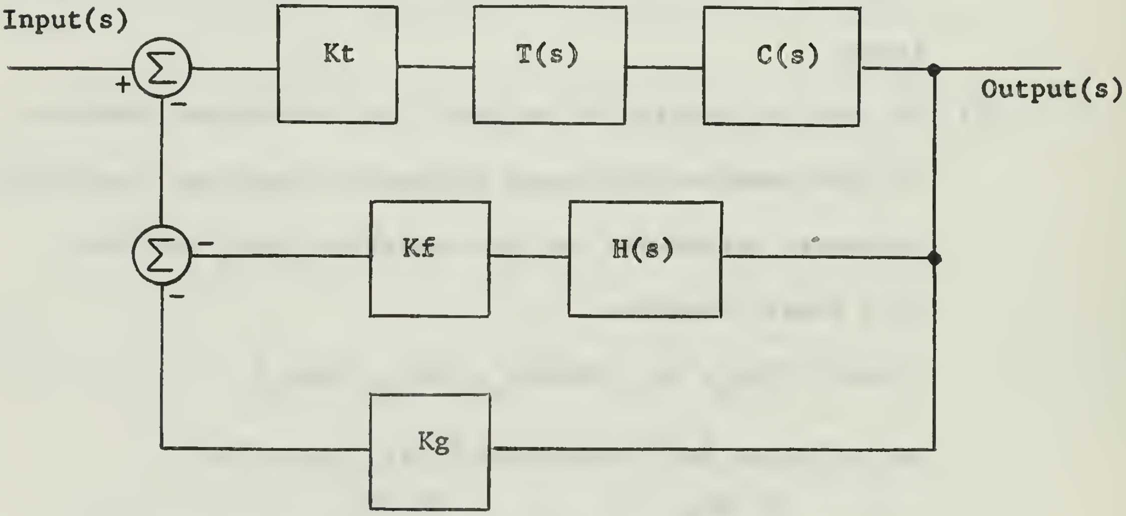

Figure 4
Control System with Two Feedback Paths

The transfer function for the control system illustrated in Figure 4 may be written

$$F(s) = \frac{Kt\ T(s) \cdot C(s)}{1 + Kt\ T(s) \cdot C(s)\ [Kf\ H(s) + Kg]} \qquad (4.3)$$

which reduces to

$$F(s) = \frac{Kt\ Tn\ Hd(s) \cdot Cn(s)}{TdHd(s) \cdot Cd(s) + KtKf\ TnHn(s) \cdot Cn(s) + KtKg\ TnHd(s) \cdot Cn(s)} \qquad (4.4)$$

The system of simultaneous equations obtained by equating coefficients of the characteristic equation will be linear in the variables (Cd_o, Cd_1, --- Cd_m), ($KtKfCn_o$, $KtKfCn_1$, --- KtKf) and ($KtKgCn_o$, $KtKgCn_1$, --- KtKg). The equations may be solved to obtain values for the variables composed of grouped parameters. However, for this system, and other multi-path feedback systems of various configurations, it has not proved possible to solve for independent values of the independent parameters. For example, in the system above $KtKfCn_o$, KtKf, $KtKgCn_o$, and KtKg are independent variables. But the value of Cn_o obtained from

$$Cn_o = \frac{KtKfCn_o}{KtKf}$$

is not necessarily the value of Cn_o obtained from

$$Cn_o = \frac{KtKgCn_o}{KtKg}$$

Systems of this type have defied solution by this method.

5. Sample Solutions

Example 5.1

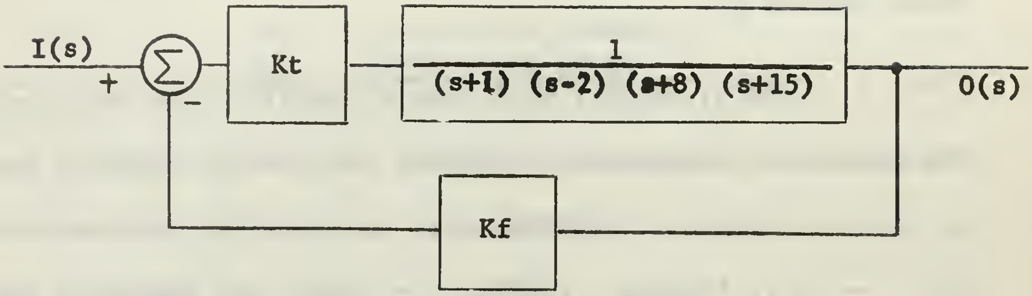

Figure 5.1
Block Diagram for Example 5.1

The open loop transfer function given above is unstable due to the pole in the right half plane ($s = 2$). Design a filter compensator and find values for forward and feedback gain which will make the system shown in Figure 5.1 absolutely stable.

Dominant roots are specified by $\xi = 0.707$ and $\omega_n = 2.5$.

No additional steady-state error or bandwidth requirements are specified.

```
                INPUT DATA
               ************


         OPEN LOOP TRANSFER FUNCTION - T(S)

     NUMERATOR OF OPEN LOOP
     TRANSFER FUNCTION - TN(S)

     TN(S)=  1.0000E+00


     DENOMINATOR OF OPEN LOOP
     TRANSFER FUNCTION - TD(S)

     NUMBER OF ROOTS = 4

     ROOT NO.       1          2          3          4

     REAL PT.  -.10E+01   .20E+01  -.80E+01  -.15E+02

     IMAG PT.   .00E+00   .00E+00   .00E+00   .00E+00

     COEFFICIENTS OF EXPANDED POLYNOMIAL

     TD(S)=  1.0000E+00 S( 4)

           + 2.2000E+01 S( 3)

           + 9.5000E+01 S( 2)

           +-1.6600E+02 S( 1)

           +-2.4000E+02

  NOTE - PROGRAM HAS READ POLE(S)
         IN THE RIGHT HALF OF THE
         COMPLEX PLANE.
```

```
          FEEDBACK PATH TRANSFER FUNCTION - H(S)

     (0) ORDER FEEDBACK IS SPECIFIED.


     NUMERATOR OF FEEDBACK PATH
     TRANSFER FUNCTION - HN(S)

     HN(S)=  1.0000E+00


     DENOMINATOR OF FEEDBACK PATH
     TRANSFER FUNCTION - HD(S)

     HD(S)=  1.0000E+00



          CLOSED LOOP TRANSFER FUNCTION
              G(S) - UNCOMPENSATED


     NUMERATOR OF CLOSED LOOP
     TRANSFER FUNCTION - GN(S)

     GN(S)= KT*  1.0000E+00


     DENOMINATOR OF CLOSED LOOP
     TRANSFER FUNCTION - GD(S)

     GD(S)=  1.0000E+00 S( 4)

           + 2.2000E+01 S( 3)

           + 9.5000E+01 S( 2)

           +-1.6600E+02 S( 1)

           + (-2.4000E+02 + KT*KX* 1.0000E+00)
```

```
            STEADY STATE ERROR SPECIFICATION

        NO STEADY STATE ERROR REQUIREMENT
        IS SPECIFIED.



            BANDWIDTH SPECIFICATION

        NO BANDWIDTH REQUIREMENT IS SPECIFIED.

     NOTE - DATA INDICATES NO BANDWIDTH OR PARAMETER
            DEPENDENT STEADY STATE ERROR SPECIFICATIONS.
            UNDER THESE CONDITIONS THE PROGRAM WILL NOT
            SOLVE FOR FORWARD GAIN AND FEEDBACK GAIN
            SEPARATELY. SOLUTION IS FOR THE PRODUCT OF
            THE GAINS.  (KT*KX)



            ROOTS SPECIFIED FOR
            CLOSED LOOP SOLUTION

        ZETA =   .7070E+00        OMEGA =   2.5000E+00

        NUMBER OF ROOTS = 2

        ROOT NO.        1          2

        REAL PT.   -.18E+01  -.18E+01

        IMAG PT.    .18E+01  -.18E+01
```

```
                  SOLUTION OF THE PROBLEM
                  ***************************

      PROGRAM REQUIRES  7 ROOTS FOR SOLUTION.


           SOLUTION OF THE SYSTEM OF LINEAR EQUATIONS

      VARIABLE             PARAMETERS          RELATIVE ERROR

      VARIABLE ( 1)          CD(0)                 7.358E-09 PERCENT

      VARIABLE ( 2)          CD(1)                  .000E-01 PERCENT

      VARIABLE ( 3)          CD(2)                  .000E-01 PERCENT

      VARIABLE ( 4)          CD(3)                -4.180E-09 PERCENT

      VARIABLE ( 5)          KT*KX*CN(0)          -1.032E-07 PERCENT

      VARIABLE ( 6)          KT*KX*CN(1)           3.466E-07 PERCENT

      VARIABLE ( 7)          KT*KX*CN(2)           1.393E-06 PERCENT

      VARIABLE ( 8)          KT*KX*CN(3)           2.421E-06 PERCENT

  NOTE - RELATIVE ERROR STATED ABOVE
         MEASURES THE ACCURACY OF THE
         SOLUTION FOR EACH VARIABLE.
```

```
          SOLUTION FOR INDIVIDUAL PARAMETERS

     NUMERATOR OF CASCADE COMPENSATION

     CN(S)=  1.0000E+00 S( 3)

           + 1.1867E+01 S( 2)

           + 3.4320E+01 S( 1)

           + 2.7105E+01

     ZEROS OF THE FILTER

     ROOT NO.       1         2          3

     REAL PT.  -.80E+01  -.13E+01  -.25E+01

     IMAG PT.   .00E+00   .00E+00   .00E+00

NOTE - ACCURACY OF ROOT FINDING SUBROUTINE
       HAS BEEN CHECKED BY EXPANDING THESE
       ROOTS, AND COMPARING THE COEFFICIENTS
       OBTAINED WITH THE COEFFICIENTS OF THE
       ORIGINAL POLYNOMIAL.
       ERROR NOTED IS LESS THAN ONE PERCENT
       FOR ALL COEFFICIENTS. SOLUTION FOR
       ROOTS IS CONSIDERED ACCURATE.
```

```
    DENOMINATOR OF CASCADE COMPENSATION

    CD(S)=  1.0000E+00 S( 3)

          + 1.5117E+01 S( 2)

          + 1.4418E+02 S( 1)

          + 2.9161E+02

    POLES OF THE FILTER

    ROOT NO.        1          2          3

    REAL PT.  -.63E+01  -.63E+01  -.26E+01

    IMAG PT.  -.85E+01   .85E+01   .00E+00

NOTE - ACCURACY HAS BEEN CHECKED,AS BEFORE.
       ERROR NOTED IS LESS THAN ONE PERCENT
       FOR ALL COEFFICIENTS. SOLUTION FOR
       ROOTS IS CONSIDERED ACCURATE.

    PRODUCT OF GAINS (KT*KX) = 5.4508E+03

    FORWARD GAIN (KT) IS  5.4508E+03,
    WITH UNITY FEEDBACK GAIN.  (KX=1.0)
```

```
             CLOSED LOOP TRANSFER FUNCTION
                   OF SOLVED PROBLEM

        NUMERATOR POLYNOMIAL

        GN(S)=  5.4508E+03 S( 3)

              + 6.4686E+04 S( 2)

              + 1.8707E+05 S( 1)

              + 1.4775E+05

        ZEROS OF THE TRANSFER FUNCTION

        ROOT NO.        1          2          3

        REAL PT.  -.80E+01  -.13E+01  -.25E+01

        IMAG PT.    .00E+00   .00E+00   .00E+00
```

NOTE - ACCURACY HAS BEEN CHECKED, AS BEFORE. ERROR NOTED IS LESS THAN ONE PERCENT FOR ALL COEFFICIENTS. SOLUTION FOR ROOTS IS CONSIDERED ACCURATE.

```
DENOMINATOR POLYNOMIAL

GD(S)=  1.0000E+00 S( 7)

      + 3.7118E+01 S( 6)

      + 5.7177E+02 S( 5)

      + 4.7338E+03 S( 4)

      + 2.2814E+04 S( 3)

      + 6.4826E+04 S( 2)

      + 1.0406E+05 S( 1)

      + 7.7761E+04

POLES OF THE SYSTEM AND ROOTS
OF THE CHARACTERISTIC EQUATION

ROOT NO.        1          2          3          4          5

REAL PT.   -.18E+01   -.18E+01   -.53E+01   -.53E+01   -.71E+01

IMAG PT.    .18E+01   -.18E+01    .18E+00   -.18E+00    .18E+00

ROOT NO.        6          7

REAL PT.   -.71E+01   -.88E+01

IMAG PT.   -.18E+00    .00E+00
```

Example 5.2

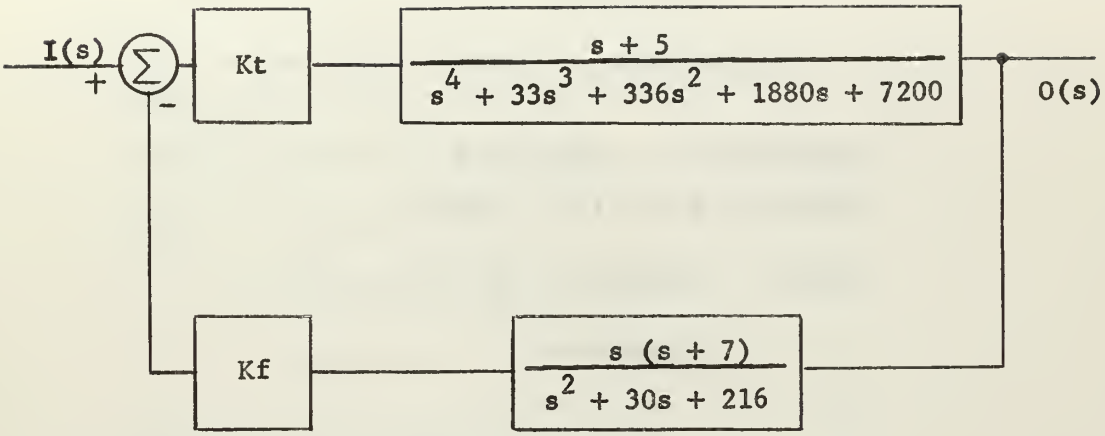
I(s)
+
−
Kt
$\frac{s + 5}{s^4 + 33s^3 + 336s^2 + 1880s + 7200}$
O(s)
Kf
$\frac{s(s + 7)}{s^2 + 30s + 216}$

Figure 5.2
Block Diagram for Example 5.2

Cascade compensation and the tachometer feedback device given are to be used in the closed loop system shown in Figure 5.2. The maximum permissible steady-state error for a step input is to be 10 percent.

Dominant roots are specified by $\xi = 0.5$ and $\omega_n = 6$.

No bandwidth requirement is specified.

```
                    INPUT DATA
                   ************


             OPEN LOOP TRANSFER FUNCTION - T(S)

     NUMERATOR OF OPEN LOOP
     TRANSFER FUNCTION - TN(S)

     TN(S)=  1.0000E+00 S( 1)

           + 5.0000E+00


     DENOMINATOR OF OPEN LOOP
     TRANSFER FUNCTION - TD(S)

     TD(S)=  1.0000E+00 S( 4)

           + 3.3000E+01 S( 3)

           + 3.3600E+02 S( 2)

           + 1.8800E+03 S( 1)

           + 7.2000E+03

     ROOTS OF TD(S)

     ROOT NO.       1          2          3          4

     REAL PT.  -.90E+01  -.20E+02  -.20E+01  -.20E+01

     IMAG PT.   .00E+00   .00E+00  -.60E+01   .60E+01
     FEEDBACK PATH TRANSFER FUNCTION - H(S)

(1) ORDER FEEDBACK IS SPECIFIED.

NUMERATOR OF FEEDBACK PATH
TRANSFER FUNCTION - HN(S)

HN(S)=  1.0000E+00 S( 2)

      + 7.0000E+00 S( 1)

      +  .0000E-01

DENOMINATOR OF FEEDBACK PATH
TRANSFER FUNCTION - HD(S)

HD(S)=  1.0000E+00 S( 2)

      + 3.0000E+01 S( 1)

      + 2.1600E+02
     CLOSED LOOP TRANSFER FUNCTION
         G(S) - UNCOMPENSATED

NUMERATOR OF CLOSED LOOP
TRANSFER FUNCTION - GN(S)

GN(S)= KT*(  1.0000E+00 S( 3)

           + 3.5000E+01 S( 2)

           + 3.6600E+02 S( 1)

           + 1.0800E+03 )


DENOMINATOR OF CLOSED LOOP
TRANSFER FUNCTION - GD(S)

GD(S)=  1.0000E+00 S( 6)

      + 6.3000E+01 S( 5)

      + 1.5420E+03 S( 4)

      + ( 1.9088E+04 + KT*KX* 1.0000E+00) S( 3)

      + ( 1.3618E+05 + KT*KX* 1.2000E+01) S( 2)

      + ( 6.2208E+05 + KT*KX* 3.5000E+01) S( 1)

      + ( 1.5552E+06 + KT*KX*  .0000E-01)
     STEADY STATE ERROR SPECIFICATION

ERROR SPECIFIED (ERROR / INPUT) IS  .1000E+00



     BANDWIDTH SPECIFICATION

NO BANDWIDTH REQUIREMENT IS SPECIFIED.



     ROOTS SPECIFIED FOR
     CLOSED LOOP SOLUTION

ZETA =  .5000E+00        OMEGA =  6.0000E+00

NUMBER OF ROOTS = 2

ROOT NO.       1         2

REAL PT.  -.30E+01  -.30E+01

IMAG PT.   .52E+01  -.52E+01
```

2.5

```
                    SOLUTION OF THE PROBLEM
                   ***************************

         PROGRAM REQUIRES 11 ROOTS FOR SOLUTION.


              STEADY STATE ERROR SOLUTION

         STEADY STATE ERROR IS   .1000E+00

         RELATIVE SPECIFICATION
         ERROR IS  3.727E-06 PERCENT

         ERROR SPECIFICATION IS SATISFIED.
```

```
            SOLUTION OF THE SYSTEM OF LINEAR EQUATIONS

    VARIABLE              PARAMETERS         RELATIVE ERROR

    VARIABLE ( 1)           CD(0)                  .000E-01 PERCENT

    VARIABLE ( 2)           CD(1)              -1.460E-09 PERCENT

    VARIABLE ( 3)           CD(2)              -3.589E-09 PERCENT

    VARIABLE ( 4)           CD(3)               1.759E-09 PERCENT

    VARIABLE ( 5)           CD(4)                  .000E-01 PERCENT

    VARIABLE ( 6)           CD(5)                  .000E-01 PERCENT

    VARIABLE ( 7)           KT*KX*CN(0)        -5.420E-09 PERCENT

    VARIABLE ( 8)           KT*KX*CN(1)        -1.270E-08 PERCENT

    VARIABLE ( 9)           KT*KX*CN(2)        -3.431E-08 PERCENT

    VARIABLE (10)           KT*KX*CN(3)         9.962E-08 PERCENT

    VARIABLE (11)           KT*KX*CN(4)         8.720E-07 PERCENT

    VARIABLE (12)           KT*KX*CN(5)         3.143E-06 PERCENT

    VARIABLE (13)           KT*CN( 0)           3.727E-06 PERCENT

NOTE - RELATIVE ERROR STATED ABOVE
       MEASURES THE ACCURACY OF THE
       SOLUTION FOR EACH VARIABLE.
```

```
     SOLUTION FOR INDIVIDUAL PARAMETERS

  NUMERATOR OF CASCADE COMPENSATION

  CN(S)=  1.0000E+00 S( 5)

        + 6.7668E+01 S( 4)

        + 1.7775E+03 S( 3)

        + 2.2639E+04 S( 2)

        + 1.3997E+05 S( 1)

        + 3.3682E+05

  ZEROS OF THE FILTER

  ROOT NO.        1          2          3          4          5

  REAL PT.  -.20E+02  -.12E+02  -.18E+02  -.84E+01  -.92E+01

  IMAG PT.   .00E+00   .00E+00   .00E+00   .00E+00   .00E+00

NOTE - ACCURACY OF ROOT FINDING SUBROUTINE
       HAS BEEN CHECKED BY EXPANDING THESE
       ROOTS, AND COMPARING THE COEFFICIENTS
       OBTAINED WITH THE COEFFICIENTS OF THE
       ORIGINAL POLYNOMIAL.
       ERROR NOTED IS LESS THAN ONE PERCENT
       FOR ALL COEFFICIENTS. SOLUTION FOR
       ROOTS IS CONSIDERED ACCURATE.
```

```
DENOMINATOR OF CASCADE COMPENSATION

CD(S)=  1.0000E+00 S( 5)

      + 7.2000E+01 S( 4)

      + 2.[illegible]591E+03 S( 3)

      + 2.7312E+04 S( 2)

      + 1.7124E+05 S( 1)

      + 4.1634E+05

POLES OF THE FILTER

ROOT NO.        1          2          3          4          5

REAL PT.  -.24E+02  -.24E+02  -.89E+01  -.78E+01  -.78E+01

IMAG PT.  -.11E+02   .11E+02   .00E+00  -.26E+01   .26E+01
```

NOTE - ACCURACY HAS BEEN CHECKED, AS BEFORE. ERROR NOTED IS LESS THAN ONE PERCENT FOR ALL COEFFICIENTS. SOLUTION FOR ROOTS IS CONSIDERED ACCURATE.

```
FORWARD GAIN (KT) = 1.6020E+03

FEEDBACK GAIN (KX) = 1.2011E+00
```

```
              CLOSED LOOP TRANSFER FUNCTION
                   OF SOLVED PROBLEM

        NUMERATOR POLYNOMIAL

        GN(S)=  1.6020E+03 S( 8)

              + 1.6447E+05 S( 7)

              + 7.2280E+06 S( 6)

              + 1.7734E+08 S( 5)

              + 2.6528E+09 S( 4)

              + 2.4737E+10 S( 3)

              + 1.4012E+11 S( 2)

              + 4.3966E+11 S( 1)

              + 5.8275E+11

        ZEROS OF THE TRANSFER FUNCTION

        ROOT NO.        1          2          3          4          5

        REAL PT.  -.20E+02   -.18E+02   -.18E+02   -.12E+02   -.12E+02

        IMAG PT.   .00E+00   -.46E-02    .46E-02   -.36E-02    .36E-02

        ROOT NO.        6          7          8

        REAL PT.  -.84E+01   -.92E+01   -.50E+01

        IMAG PT.   .00E+00    .00E+00    .00E+00

     NOTE - ACCURACY HAS BEEN CHECKED,AS BEFORE.
            ERROR NOTED IS LESS THAN ONE PERCENT
            FOR ALL COEFFICIENTS. SOLUTION FOR
            ROOTS IS CONSIDERED ACCURATE.
```

```
DENOMINATOR POLYNOMIAL

GD(S)=  1.0000E+00 S(11)

      + 1.3500E+02 S(10)

      + 8.1371E+03 S( 9)

      + 2.8907E+05 S( 8)

      + 6.7308E+06 S( 7)

      + 1.0810E+08 S( 6)

      + 1.2275E+09 S( 5)

      + 9.9345E+09 S( 4)

      + 5.6863E+10 S( 3)

      + 2.2290E+11 S( 2)

      + 5.4799E+11 S( 1)

      + 6.4750E+11

POLES OF THE SYSTEM AND ROOTS
OF THE CHARACTERISTIC EQUATION

ROOT NO.       1          2          3          4          5

REAL PT.  -.30E+01   -.30E+01   -.90E+01   -.90E+01   -.12E+02

IMAG PT.   .52E+01   -.52E+01    .52E+00   -.52E+00    .52E+00

ROOT NO.       6          7          8          9         10

REAL PT.  -.12E+02   -.15E+02   -.15E+02   -.18E+02   -.18E+02

IMAG PT.  -.52E+00    .52E+00   -.52E+00    .52E+00   -.52E+00

ROOT NO.      11

REAL PT.  -.21E+02

IMAG PT.   .00E+00
```

Example 5.3

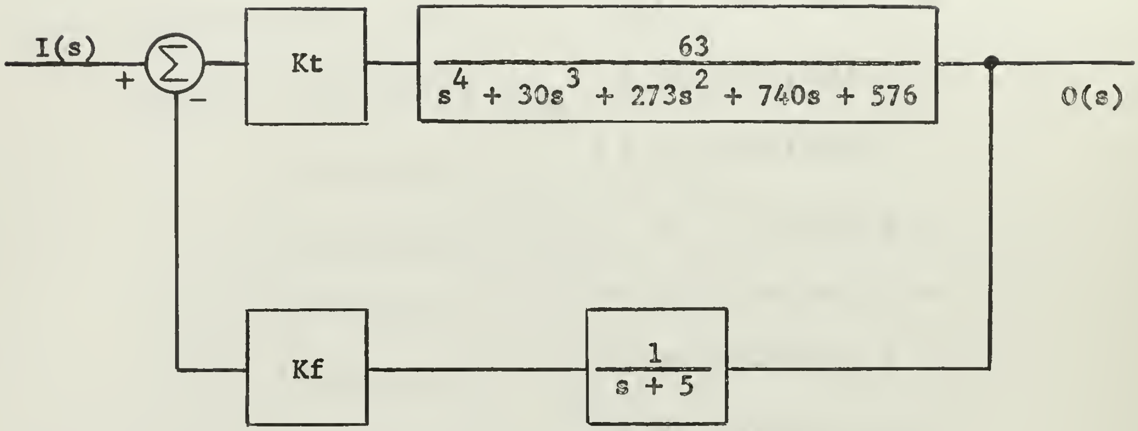

Figure 5.3
Block Diagram for Example 5.3

The gain of the open loop device shown in Figure 5.3 is less than -20 db for all frequencies. Design a filter compensator and find values of forward and feedback gain which will provide positive gain at low frequencies and a bandwidth of zero db at 30 radians per second.

Dominant roots are specified by $\xi = 0.5$ and $\omega_n = 10$.

No steady-state error requirement is specified.

```
            INPUT DATA
           ************


      OPEN LOOP TRANSFER FUNCTION - T(S)

NUMERATOR OF OPEN LOOP
TRANSFER FUNCTION - TN(S)

TN(S)=  6.3[illegible]0E+01


DENOMINATOR OF OPEN LOOP
TRANSFER FUNCTION - TD(S)

TD(S)=  1.[illegible]E+00 S( 4)

      + 3.[illegible]E+01 S( 3)

      + 2.73[illegible]E+02 S( 2)

      + 7.4[illegible]0E+02 S( 1)

      + 5.76[illegible]E+02

ROOTS OF TD(S)

ROOT NO.        1         2         3         4

REAL PT.  -.12E+02  -.15E+02  -.14E+01  -.25E+01

IMAG PT.   .00E+00   .00E+00   .00E+00   .00E+00
```

```
    FEEDBACK PATH TRANSFER FUNCTION - H(S)

(0) ORDER FEEDBACK IS SPECIFIED.


NUMERATOR OF FEEDBACK PATH
TRANSFER FUNCTION - HN(S)

HN(S)=  1.0000E+00


DENOMINATOR OF FEEDBACK PATH
TRANSFER FUNCTION - HD(S)

HD(S)=  1.0000E+00 S( 1)

      + 5.0000E+00
```

```
     CLOSED LOOP TRANSFER FUNCTION
         G(S) - UNCOMPENSATED

NUMERATOR OF CLOSED LOOP
TRANSFER FUNCTION - GN(S)

GN(S)= KT*(  6.3000E+01 S( 1)

           + 3.1500E+02 )


DENOMINATOR OF CLOSED LOOP
TRANSFER FUNCTION - GD(S)

GD(S)=  1.0000E+00 S( 5)

      + 3.5000E+01 S( 4)

      + 4.2300E+02 S( 3)

      + 2.1150E+03 S( 2)

      + 4.2760E+03 S( 1)

      + ( 2.8800E+03 + KT*KX* 6.3000E+01)
     STEADY STATE ERROR SPECIFICATION

NO STEADY STATE ERROR REQUIREMENT
IS SPECIFIED.


     BANDWIDTH SPECIFICATION

GAIN OF   .000E-01 DB  AT FREQUENCY
OF  3.000E+01 RAD/SEC IS SPECIFIED.


     ROOTS SPECIFIED FOR
     CLOSED LOOP SOLUTION

ZETA =  .5000E+00        OMEGA =  1.0000E+01

NUMBER OF ROOTS = 2

ROOT NO.       1          2

REAL PT.  -.50E+01  -.50E+01

IMAG PT.   .87E+01  -.87E+01
```

```
                 SOLUTION OF THE PROBLEM
                ****************************


PROGRAM REQUIRES  9 ROOTS FOR SOLUTION.



     BANDWIDTH SOLUTION

GAIN IS    .0000E-01 DB  AT FREQUENCY 3.000E+01 RAD/SEC

RELATIVE ERROR IS   .0000E-01 PERCENT

BANDWIDTH SPECIFICATION IS SATISFIED.
```

```
                 THE SYSTEM OF LINEAR EQUATIONS

                  PARAMETERS        RELATIVE ERROR

VARIABLE  ( 1)     CD(0)               -2.355E-09 PERCENT

VARIABLE ( 2)      CD(1)               -5.282E-09 PERCENT

VARIABLF ( 3)      CD(2)               -7.561E-08 PERCENT

VARIABLE ( 4)      CD(3)               -3.551E-07 PERCENT

VARIABLE ( 5)      CD(4)                1.559E-06 PERCENT

VARIABLE ( 6)      KT*KX*CN(0)          7.654E-05 PERCENT

VARIABLE ( 7)      KT*KX*CN(1)          1.068E-03 PERCENT

VARIABLE ( 8)      KT*KX*CN(2)          8.962E-03 PERCENT

VARIABLF ( 9)      KT*KX*CN(3)          5.289E-02 PERCENT

VARIABLE (10)      KT*KX*CN(4)          2.441E-01 PERCENT

NOTE - RELATIVE ERROR STATED ABOVE
       MEASURES THE ACCURACY OF THE
       SOLUTION FOR EACH VARIABLE.
```

```
        SOLUTION FOR INDIVIDUAL PARAMETERS

   NUMERATOR OF CASCADE COMPENSATION

   CN(S)=  1.0000E+00 S( 4)

         + 3.85[illegible]7E+01 S( 3)

         + 5.4214E+02 S( 2)

         + 3.4034E+03 S( 1)

         + 8.5495E+03

   ZEROS OF THE FILTER

   ROOT NO.         1          2          3          4

   REAL PT.  -.12E+02  -.15E+02  -.61E+01  -.61E+01

   IMAG PT.   .00E+00   .00E+00  -.36E+01   .36E+01

NOTE - ACCURACY OF ROOT FINDING SUBROUTINE
       HAS BEEN CHECKED BY EXPANDING THESE
       ROOTS, AND COMPARING THE COEFFICIENTS
       OBTAINED WITH THE COEFFICIENTS OF THE
       ORIGINAL POLYNOMIAL.
       ERROR NOTED IS LESS THAN ONE PERCENT
       FOR ALL COEFFICIENTS. SOLUTION FOR
       ROOTS IS CONSIDERED ACCURATE.
```

```
    DENOMINATOR OF CASCADE COMPENSATION

    CD(S)=  1.0024E+00 S( 4)

          + 1.2500E+02 S( 3)

          + 6.3543E+03 S( 2)

          + 1.6789E+05 S( 1)

          + 2.4427E+06

    POLES OF THE FILTER

    ROOT NO.        1          2          3          4

    REAL PT.  -.47E+02  -.47E+02  -.15E+02  -.15E+02

    IMAG PT.  -.17E+02   .17E+02  -.27E+02   .27E+02

NOTE - ACCURACY HAS BEEN CHECKED,AS BEFORE.
       ERROR NCTED IS LESS THAN ONE PERCENT
       FOR ALL COEFFICIENTS. SOLUTION FOR
       ROOTS IS CONSIDERED ACCURATE.

    FORWARD GAIN (KT) = 4.2411E+04

    FEEDBACK GAIN (KX) = 7.1267E+00
```

```
          CLOSED LOOP TRANSFER FUNCTION
                 OF SOLVED PROBLEM

      NUMERATOR POLYNOMIAL

      CN(S)=   2.6719E+06 S( 5)

           + 1.1625E+08 S( 4)

           + 1.963 E+09 S( 3)

           + 1.6336E+10 S( 2)

           + 6.9311E+10 S( 1)

           + 1.1422E+11

      ZEROS OF THE TRANSFER FUNCTION

      ROOT NO.         1          2          3          4          5

      REAL PT.  -.61E+01  -.61E+01  -.15E+02  -.12E+02  -.50E+01

      IMAG PT.  -.36E+01   .36E+01   .00E+00   .00E+00   .00E+00

   NOTE - ACCURACY HAS BEEN CHECKED,AS BEFORE.
          ERROR NOTED IS GREATER THAN ONE PERCENT
          FOR AT LEAST ONE COEFFICIENT. SOLUTION
          FOR ROOTS IS NOT CONSIDERED ACCURATE.
```

```
DENOMINATOR POLYNOMIAL

GD(S)=  1.0000E+00 S( 9)

      + 1.6000E+02 S( 8)

      + 1.1152E+04 S( 7)

      + 4.4527E+05 S( 6)

      + 1.1274E+07 S( 5)

      + 1.8947E+08 S( 4)

      + 2.1474E+09 S( 3)

      + 1.6201E+10 S( 2)

      + 7.5736E+10 S( 1)

      + 1.6983E+11

POLES OF THE SYSTEM AND ROOTS
OF THE CHARACTERISTIC EQUATION

ROOT NO.        1          2          3          4          5

REAL PT.  -.50E+01   -.50E+01   -.15E+02   -.15E+02   -.20E+02

IMAG PT.    .87E+01   -.87E+01    .87E+00   -.87E+00    .87E+00

ROOT NO.        6          7          8          9

REAL PT.  -.20E+02   -.25E+02   -.25E+02   -.30E+02

IMAG PT.  -.87E+00    .87E+00   -.87E+00    .00E+00
```

Example 5.4

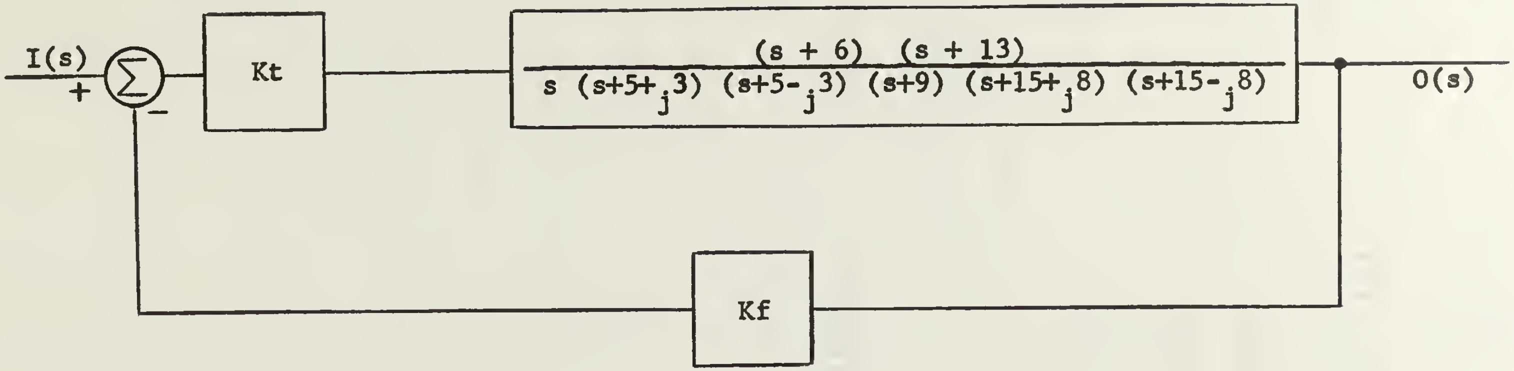

Figure 5.4
Block Diagram for Example 5.4

Design a filter compensator and find values for forward and feedback gain which will give the system shown in Figure 5.4 a bandwidth of -3 db at 20 radians per second. Maximum permissible steady-state error for a step input is 10 percent.

Dominant roots are specified by $\xi = 0.707$ and $\omega_n = 4$.

```
          INPUT DATA
         ************


     OPEN LOCP TRANSFER FUNCTION - T(S)

NUMERATOR OF OPEN LOOP
TRANSFER FUNCTION - TN(S)

NUMBER OF ROOTS = 2

ROOT NO.        1         2

REAL PT.   -.60E+01  -.13E+02

IMAG PT.    .00E+00   .00E+00

COEFFICIENTS OF EXPANDED POLYNOMIAL

TN(S)=  1.0000E+00 S( 2)

      + 1.9000E+01 S( 1)

      + 7.8000E+01
```

```
DENOMINATOR OF OPEN LOOP
TRANSFER FUNCTION - TD(S)

NUMBER OF ROOTS = 6

ROOT NO.         1          2          3          4          5

REAL PT.   .00E+00  -.50E+01  -.50E+01  -.90E+01  -.15E+02

IMAG PT.   .00E+00   .30E+01  -.30E+01   .00E+00   .80E+01

ROOT NO.         6

REAL PT.  -.15E+02

IMAG PT.  -.80E+01

COEFFICIENTS OF EXPANDED POLYNOMIAL

TD(S)=  1.0000E+00 S( 6)

      + 4.9000E+01 S( 5)

      + 9.8300E+02 S( 4)

      + 9.5170E+03 S( 3)

      + 4.5016E+04 S( 2)

      + 8.8434E+04 S( 1)

      +  .0000E-01
```

```
     FEEDBACK PATH TRANSFER FUNCTION - H(S)

(0) ORDER FEEDBACK IS SPECIFIED.


NUMERATOR OF FEEDBACK PATH
TRANSFER FUNCTION - HN(S)

HN(S)=  1.0000E+00


DENOMINATOR OF FEEDBACK PATH
TRANSFER FUNCTION - HD(S)

HD(S)=  1.0000E+00
```

```
     CLOSED LOOP TRANSFER FUNCTION
          G(S) - UNCOMPENSATED

NUMERATOR OF CLOSED LOOP
TRANSFER FUNCTION - GN(S)

GN(S)= KT*(  1.0000E+00 S( 2)

           + 1.9000E+01 S( 1)

           + 7.8000E+01 )


DENOMINATOR OF CLOSED LOOP
TRANSFER FUNCTION - GD(S)

GD(S)=  1.0000E+00 S( 6)

      + 4.9000E+01 S( 5)

      + 9.8350E+02 S( 4)

      + 9.5170E+03 S( 3)

      + ( 4.5016E+04 + KT*KX* 1.0000E+00) S( 2)

      + ( 8.8434E+04 + KT*KX* 1.9000E+01) S( 1)

      + (   .0000E-01 + KT*KX* 7.8000E+01)
     STEADY STATE ERROR SPECIFICATION

ERROR SPECIFIED (ERROR / INPUT) IS  .1000E+00



     BANDWIDTH SPECIFICATION

GAIN OF -3.000E+00 DB  AT FREQUENCY
OF  2.000E+01 RAD/SEC IS SPECIFIED.



     ROOTS SPECIFIED FOR
     CLOSED LOOP SOLUTION

ZETA =  .7070E+00       OMEGA =  4.0000E+00

NUMBER OF ROOTS = 2

ROOT NO.       1         2

REAL PT.  -.28E+01  -.28E+01

IMAG PT.   .28E+01  -.28E+01
```

```
          SOLUTION OF THE PROBLEM
         **************************


PROGRAM REQUIRES 11 ROOTS FOR SOLUTION.



     BANDWIDTH SOLUTION

ITERATIVE PROCEDURE FOR BANDWIDTH SOLUTION

PROCESS IS CONVERGING

ITERATION     GAIN-DB    RELATIVE ERROR

    31        -2.64E+00    1.20E+01 PERCENT

    36        -3.02E+00   -6.65E-01 PERCENT

GAIN IS -3.020E+00 DB  AT FREQUENCY 2.000E+01 RAD/SEC

RELATIVE ERROR IS -6.655E-01 PERCENT

BANDWIDTH SPECIFICATION IS SATISFIED.



     STEADY STATE ERROR SOLUTION

STEADY STATE ERROR IS  .1000E+00

RELATIVE SPECIFICATION
ERROR IS  3.274E-08 PERCENT

ERROR SPECIFICATION IS SATISFIED.
```

```
          SOLUTION OF THE SYSTEM OF LINEAR EQUATIONS

      VARIABLE              PARAMETERS        RELATIVE ERROR

      VARIABLE ( 1)           CD(0)            -2.290E-09 PERCENT

      VARIABLE ( 2)           CD(1)               .000E-01 PERCENT

      VARIABLE ( 3)           CD(2)               .000E-01 PERCENT

      VARIABLE ( 4)           CD(3)              1.689E-09 PERCENT

      VARIABLE ( 5)           CD(4)               .000E-01 PERCENT

      VARIABLE ( 6)           CD(5)              3.595E-09 PERCENT

      VARIABLE ( 7)           KT*KX*CN(0)        1.032E-08 PERCENT

      VARIABLE ( 8)           KT*KX*CN(1)        1.033E-07 PERCENT

      VARIABLE ( 9)           KT*KX*CN(2)        1.555E-06 PERCENT

      VARIABLE (10)           KT*KX*CN(3)        1.347E-05 PERCENT

      VARIABLE (11)           KT*KX*CN(4)        8.095E-05 PERCENT

      VARIABLE (12)           KT*KX*CN(5)        4.292E-04 PERCENT

      VARIABLE (13)           KT*CN( 0)         -5.120E+09 PERCENT

   NOTE - RELATIVE ERROR STATED ABOVE
          MEASURES THE ACCURACY OF THE
          SOLUTION FOR EACH VARIABLE.
```

```
        SOLUTION FOR INDIVIDUAL PARAMETERS

   NUMERATOR OF CASCADE COMPENSATION

   CN(S)=  1.0000E+00 S( 5)

         + 4.5730E+01 S( 4)

         + 8.3424E+02 S( 3)

         + 6.9979E+03 S( 2)

         + 2.7515E+04 S( 1)

         + 4.7798E+04

   ZEROS OF THE FILTER

   ROOT NO.        1          2          3          4          5

   REAL PT.  -.15E+02   -.15E+02   -.90E+01   -.34E+01   -.34E+01

   IMAG PT.  -.76E+01    .76E+01    .00E+00   -.27E+01    .27E+01


NOTE - ACCURACY OF ROOT FINDING SUBROUTINE
       HAS BEEN CHECKED BY EXPANDING THESE
       ROOTS, AND COMPARING THE COEFFICIENTS
       OBTAINED WITH THE COEFFICIENTS OF THE
       ORIGINAL POLYNOMIAL.
       ERROR NOTED IS LESS THAN ONE PERCENT
       FOR ALL COEFFICIENTS. SOLUTION FOR
       ROOTS IS CONSIDERED ACCURATE.
```

```
DENOMINATOR OF CASCADE COMPENSATION

CD(S)=  1.0000E+00 S( 5)

      + 1.7057E+02 S( 4)

      + 1.1647E+04 S( 3)

      + 4.0007E+05 S( 2)

      + 4.0045E+06 S( 1)

      + 5.8576E+06

POLES OF THF FILTER

ROOT NO.        1          2          3          4          5

REAL PT.  -.80E+02  -.38E+02  -.38E+02  -.18E+01  -.13E+02

IMAG PT.   .00E+00  -.43E+02   .43E+02   .00E+00   .00E+00


NOTE - ACCURACY HAS BEEN CHECKED,AS BEFORE.
       ERROR NOTED IS LESS THAN ONE PERCENT
       FOR ALL COEFFICIENTS. SOLUTION FOR
       ROOTS IS CONSIDERED ACCURATE.

FORWARD GAIN (KT) = 2.6980E+06

FEEDBACK GAIN (KX) = 1.1111E+00
```

```
     CLOSED LOOP TRANSFER FUNCTION
           OF SOLVED PROBLEM

  NUMERATOR POLYNOMIAL

  GN(S)=   2.6980E+06 S( 7)

         + 1.7464E+08 S( 6)

         + 4.8055E+09 S( 5)

         + 7.1269E+10 S( 4)

         + 6.7853E+11 S( 3)

         + 3.0121E+12 S( 2)

         + 8.2407E+12 S( 1)

         + 1.0059E+13

  ZEROS OF THE TRANSFER FUNCTION

  ROOT NO.        1          2          3          4          5

  REAL PT.   -.13E+02   -.15E+02   -.15E+02   -.60E+01   -.90E+01

  IMAG PT.    .00E+00   -.76E+01    .76E+01    .00E+00    .00E+00

  ROOT NO.        6          7

  REAL PT.   -.34E+01   -.34E+01

  IMAG PT.   -.27E+01    .27E+01

NOTE - ACCURACY HAS BEEN CHECKED,AS BEFORE.
       ERROR NOTED IS LESS THAN ONE PERCENT
       FOR ALL COEFFICIENTS. SOLUTION FOR
       ROOTS IS CONSIDERED ACCURATE.
```

```
DENOMINATOR POLYNOMIAL

GD(S)=  1.0000E+00 S(11)

      + 2.1957E+02 S(10)

      + 2.0988E+04 S( 9)

      + 1.1480E+06 S( 8)

      + 3.9723E+07 S( 7)

      + 9.0800E+08 S( 6)

      + 1.3910E+10 S( 5)

      + 1.4210E+11 S( 4)

      + 9.4753E+11 S( 3)

      + 3.9646E+12 S( 2)

      + 9.6743E+12 S( 1)

      + 1.1177E+13

POLES OF THE SYSTEM AND ROOTS
OF THE CHARACTERISTIC EQUATION

ROOT NO.        1          2          3          4          5

REAL PT.  -.28E+01  -.28E+01  -.85E+01  -.85E+01  -.17E+02

IMAG PT.   .28E+01  -.28E+01   .28E+00  -.28E+00   .28E+00

ROOT NO.        6          7          8          9         10

REAL PT.  -.17E+02  -.26E+02  -.26E+02  -.34E+02  -.34E+02

IMAG PT.  -.28E+00   .28E+00  -.28E+00   .28E+00  -.28E+00

ROOT NO.       11

REAL PT.  -.43E+02

IMAG PT.   .00E+00
```

6. Conclusion

The method presented has been shown to be fast and accurate, and capable of designing systems which will meet specifications for root dominance, steady-state error, and bandwidth.

The principal disadvantage of the method presented, exclusive of the restrictions mentioned in Section 4, is the large number of filter sections required, particularly for higher order open loop systems. The filter sections tend to further complicate the originally complex systems.

The possibility of reducing the number of filter sections employed in the final design is worthy of further study. Approximation of the characteristics of the cascade compensation may be accomplished with a filter of lower order, but analysis of the effects of such approximation on system specifications is beyond the scope of this paper.

Bibliography

1. Thaler, G. J. and Brown, R. G. Analysis and Design of Feedback Control Systems. Mc Graw-Hill Book Company, Inc., 1960.

2. Thaler, G. J., Siljak, D. D. and Dorf, R. C. Algebraic Methods for Dynamic Systems. Interim Technical Report prepared for NASA under Contract NAS2-2609, 1966.

APPENDIX

PROGRAM COMPTECH

```
C      PROGRAM COMPTECH IS WRITTEN IN THE FORTRAN 60 LANGUAGE FOR USE
C      WITH A CDC 1604 COMPUTER SYSTEM. THE TAPE LIBRARY AVAILABLE AT
C      NAVAL POSTGRADUATE SCHOOL COMPUTER FACILITY PERMITS USE OF THE
C      SUBSCRIPT ZERO (0) TO INDICATE THE FIRST ELEMENT OF LINEAR ARRAYS.
C      THIS FEATURE IS USED EXTENSIVELY THROUGHOUT THE PROGRAM, AS THE
C      RESULTANT NOTATION IS CONSISTENT WITH THE NOTATION USED IN DEVEL-
C      OPEMENT OF THE METHOD PRESENTED IN THIS PAPER. THIS FEATURE MUST
C      BE CONSIDERED IN TRANSLATION OF THE PROGRAM INTO OTHER FORTRAN
C      LANGUAGES.
C
C
C                        INPUT DATA FORMAT
C
C      THE PROGRAM REQUIRES SPECIFICATION OF AN OPEN LOOP (FORWARD)
C      TRANSFER FUNCTION T(S), A FEEDBACK TRANSFER FUNCTION H(S), AND ONE
C      OR MORE DOMINANT ROOTS, UPON WHICH TO BASE A CLOSED LOOP DESIGN.
C      ADDITIONAL STEADY STATE ERROR (FOR A STEP INPUT) AND BANDWIDTH
C      REQUIREMENTS MAY BE SPECIFIED.
C
C
C                  OPEN LOOP TRANSFER FUNCTION T(S)
C
C      THE NUMERATOR TN(S) AND DENOMINATOR TD(S) OF THE FORWARD TRANSFER
C      FUNCTION MAY BE SPECIFIED EITHER BY ROOTS OR BY COEFFICIENTS OF A
C      POLYNOMIAL.
C      IF ROOTS ARE SPECIFIED,
C
C           FIRST DATA CARD
C              FOUR FIELDS OF I5 FORMAT
C
C                FIRST FIELD - THE INTEGER ZERO (0). INDICATES THAT ROOTS
```

```
C                             ARE TO BE READ.
C
C               SECOND FIELD - THE NUMBER OF ROOTS TO BE READ FOR TN(S).
C                              MAXIMUM IS 9.
C
C               THIRD FIELD - THE NUMBER OF ROOTS TO BE READ FOR TD(S).
C                             MAXIMUM IS 10.
C
C               FOURTH FIELD - THE INTEGER ZERO (0) IF NO ROOTS OF TN(S)
C                              OR TD(S) ARE LOCATED IN THE RIGHT HALF
C                              PLANE, OR
C
C                            - THE INTEGER ONE (1) IF ROOTS ARE LOCATED
C                              IN THE RIGHT HALF PLANE.
C
C          SECOND DATA CARD(S) - ROOTS OF TN(S)
C             EIGHT FIELDS OF E10.4 FORMAT PER CARD
C
C               EACH ROOT IS ALLOTED 20 COLUMNS OR TWO FIELDS OF TEN
C               SPACES. THE FIRST FIELD OF TEN SPACES IS ALLOTED TO THE
C               REAL PART OF THE ROOT, WHICH MUST BE A NEGATIVE VALUE IF
C               THE ROOT IS LOCATED IN THE LEFT HALF PLANE. THE SECOND
C               FIELD OF TEN SPACES IS ALLOTED TO THE IMAGINARY PART OF
C               THE ROOT, AND MUST BE LEFT BLANK IF THE ROOT IS NOT COM-
C               PLEX. ALL COMPLEX ROOTS INTRODUCED MUST HAVE CONJUGATES.
C               IF TN(S) IS A CONSTANT, THIS VALUE IS PLACED IN THE
C               FIRST FIELD OF TEN SPACES AND THE REMAINDER OF THE CARD
C               IS LEFT BLANK. EACH CARD HAS SPACE FOR FOUR ROOTS, ADDI-
C               TIONAL ROOTS REQUIRE ADDITIONAL DATA CARDS.
C
C          THIRD DATA CARD(S) - ROOTS OF TD(S)
C             SAME FORMAT AS DATA CARDS FOR ROOTS OF TN(S).
C
C
C     IF COEFFICIENTS OF POLYNOMIALS ARE SPECIFIED,
C
```

```
C          FIRST DATA CARD
C             FOUR FIELDS OF I5 FORMAT
C
C               FIRST FIELD - THE INTEGER ONE (1). INDICATES THAT
C                             COEFFICIENTS ARE TO BE READ.
C
C               SECOND FIELD - THE NUMBER OF COEFFICIENTS TO BE READ FOR
C                              TN(S). MAXIMUM IS 10.
C
C               THIRD FIELD - THE NUMBER OF COEFFICIENTS TO BE READ FOR
C                             TD(S). MAXIMUM IS 11.
C
C               FOURTH FIELD - PROGRAM WILL CHECK THE POLYNOMIALS FOR
C                              ROOTS LOCATED IN THE RIGHT HALF PLANE.
C                              IF THE VALUE SPECIFIED IN THE FOURTH
C                              FIELD IS ZERO OR BLANK, THE PROGRAM WILL
C                              NOT PROCESS RIGHT HALF PLANE ROOTS. IF
C                              ROOTS OF EITHER TN(S) OR TD(S) ARE KNOWN
C                              OR SUSPECTED TO BE IN THE RHP, THE INTEG-
C                              ER ONE (1) MUST BE ENTERED IN THE FOURTH
C                              FIELD TO AVOID AN ERROR PRINTOUT AND
C                              PROGRAM TERMINATION.
C
C          SECOND DATA CARD(S) - COEFFICIENTS OF TN(S)
C             EIGHT FIELDS OF E10.4 FORMAT
C
C               EACH COEFFICIENT IS ALLOTED 10 COLUMNS. THE PROGRAM
C               READS THE COEFFICIENTS IN ASCENDING ORDER, BEGINNING
C               WITH THE ZERO POWER OR CONSTANT TERM. FIELDS MUST BE
C               LEFT BLANK FOR COEFFICIENTS OF VALUE ZERO. IF TN(S) IS
C               A CONSTANT, THIS VALUE IS PLACED IN THE FIRST FIELD OF
C               TEN SPACES, AND THE REMAINDER OF THE CARD IS LEFT BLANK.
C               EACH CARD HAS SPACE FOR EIGHT COEFFICIENTS. ADDITIONAL
C               COEFFICIENTS REQUIRE ADDITIONAL DATA CARDS.
C
C          THIRD DATA CARDS - COEFFICIENTS OF TD(S)
```

```
C
C        SAME FORMAT AS DATA CARDS FOR TN(S), WITH THE EXCEPTION
C        THAT TD(S) MUST BE A RATIONALIZED POLYNOMIAL,I.E. THE LAST
C        COEFFICIENT READ MUST BE ONE (1.0).
C
C
C           FEEDBACK TRANSFER FUNCTION H(S)
C
C    IF TACHOMETER OR HIGHER DERIVATIVE FEEDBACK IS SPECIFIED, THIS
C    FACTOR IS READ SEPARATELY FROM THE TRANSFER FUNCTION OF THE FEED-
C    BACK DEVICE EMPLOYED. HN(S) AND HD(S) ARE THE NUMERATOR AND DENOM-
C    INATOR POLYNOMIALS OF THE FEEDBACK DEVICE TRANSFER FUNCTION,
C    EXCLUSIVE OF HIGHER ORDER FEEDBACK. THE PROGRAM WILL NOT PROCESS
C    INTEGRAL FEEDBACK.
C
C       FOURTH DATA CARD
C         THREE FIELDS OF I5 FORMAT
C
C          FIRST FIELD - ORDER OF FEEDBACK, ZERO (0) FOR DIRECT
C                        FEEDBACK, ONE (1) FOR FIRST DERIVATIVE
C                        FEEDBACK, ETC.
C
C          SECOND FIELD - THE NUMBER OF COEFFICIENTS TO BE READ FOR
C                         HN(S), EXCLUSIVE OF DERIVATIVE FEEDBACK.
C                         HN(0) MAY NOT BE ZERO. THE MAXIMUM NUMBER
C                         OF COEFFICIENTS READ IS 3.
C
C          THIRD FIELD - THE NUMBER OF COEFFICIENTS OF HD(S). THE
C                        MAXIMUM READ IS 3.
C
C       FIFTH DATA CARD - COEFFICIENTS OF HN(S)
C         SAME FORMAT AS DATA CARDS FOR COEFFICIENTS OF TN(S).
C
C       SIXTH DATA CARD - COEFFICIENTS OF HD(S)
C         SAME FORMAT AS DATA CARDS FOR COEFFICIENTS OF TN(S).
C
```

```
C     AFTER READING BOTH FORWARD AND FEEDBACK TRANSFER FUNCTIONS, THE
C     PROGRAM WILL CHECK THE INPUT AGAINST THE FOLLOWING REQUIREMENTS -
C
C        1) THE ORDER OF THE PRODUCT OF OPEN LOOP AND FEEDBACK NUMERATOR
C           POLYNOMIALS, INCLUDING DERIVATIVE FEEDBACK, MUST BE LESS
C           THAN THE ORDER OF THE PRODUCT OF THE CORRESPONDING DENOMIN-
C           ATORS.
C
C        2) THE MAXIMUM ORDER OF THIS PRODUCT OF DENOMINATORS IS 11.
C
C     IF THESE REQUIREMENTS ARE NOT SATISFIED, PROGRAM WILL PRINT OUT AN
C     ERROR STATEMENT AND TERMINATE.
C
C
C                  STEADY STATE ERROR SPECIFICATION
C
C     STEADY STATE ERROR IS SPECIFIED AS THE DECIMAL FRACTION OF THE
C     INPUT SIGNAL WHICH IS PERMISSIBLE AS SYSTEM ERROR.
C
C          SEVENTH DATA CARD
C             FIELD OF I5 FORMAT, FOLLOWED CONSECUTIVELY BY A FIELD OF
C             E10.4 FORMAT
C
C               FIRST FIELD - THE INTEGER ONE (1). INDICATES THAT AN
C                             ERROR REQUIREMENT IS SPECIFIED.
C
C               SECOND FIELD - VALUE OF STEADY STATE ERROR SPECIFIED.
C
C          IF NO STEADY STATE ERROR REQUIREMENT IS SPECIFIED, THE SEVEN-
C          TH DATA CARD MUST BE BLANK.
C
C
C                  BANDWIDTH SPECIFICATION
C
C     BANDWIDTH IS SPECIFIED AS A FREQUENCY IN RADIANS PER SECOND AT
C     WHICH THE SYSTEM WILL HAVE A DESIRED GAIN IN DECIBELS.
```

```
C
C      EIGHTH DATA CARD
C        TWO FIELDS OF E10.4 FORMAT
C
C           FIRST FIELD - GAIN DESIRED IN DB.
C
C           SECOND FIELD - BANDWIDTH FREQUENCY IN RAD/SEC.
C
C      IF NO BANDWIDTH REQUIREMENT IS SPECIFIED, THE EIGHTH DATA
C      CARD MUST BE BLANK.
C
C
C             ROOT DOMINANCE SPECIFICATION
C
C   AT LEAST ONE NON-ZERO ROOT OF THE CLOSED LOOP SYSTEM MUST BE SPEC-
C   IFIED IN ORDER TO ESTABLISH A ROOT DOMINANCE CRITERION. ANY NUM-
C   BER OF ROOTS MAY BE SPECIFIED. THE PROGRAM WILL COMPARE THE NUMBER
C   OF ROOTS SPECIFIED AGAINST THE NUMBER OF ROOTS REQUIRED FOR SOLU-
C   TION, AND PRINT OUT ANY ERROR. A PAIR OF ROOTS MAY BE DESIGNATED
C   BY DAMPING RATIO (ZETA) AND NATURAL FREQUENCY (OMEGAN).
C
C      NINTH DATA CARD
C        TWO FIELDS OF I5 FORMAT
C
C           FIRST FIELD - THE INTEGER ZERO (0) IF TWO ROOTS ARE
C                         DESIGNATED BY ZETA AND OMEGAN, OR
C
C                       - THE INTEGER ONE (1) IF NO ROOTS ARE DESIG-
C                         NATED BY ZETA AND OMEGAN.
C
C           SECOND FIELD - THE TOTAL NUMBER OF ROOTS SPECIFIED,
C                          INCLUDING THE TWO SPECIFIED BY ZETA AND
C                          OMEGAN, IF APPLICABLE.
C
C      TENTH DATA CARD - ZETA AND OMEGAN
C        TWO FIELDS OF E10.4 FORMAT
```

```
C
C                 FIRST FIELD - THE VALUE OF ZETA SPECIFIED, MUST BE LESS
C                               THAN ONE.
C
C                 SECOND FIELD - THE VALUE OF OMEGAN SPECIFIED.
C
C            IF NO ROOTS ARE SPECIFIED BY ZETA AND OMEGAN, THE TENTH DATA
C            CARD MUST BE OMITTED.
C
C            ELEVENTH DATA CARD(S) - ROOTS SPECIFIED
C
C               VALUES OF ROOTS SPECIFIED, EXCLUDING ANY SPECIFIED BY ZETA
C               AND OMEGAN. SAME FORMAT AS DATA CARDS FOR ROOTS OF TN(S).
C               ALL ROOTS SPECIFIED MUST BE LOCATED IN THE LEFT HALF PLANE
C               I.E. ALL MUST HAVE NEGATIVE REAL PARTS.
C
C
      PROGRAM COMPTECH
      DIMENSION TNR(10),TNI(10),TDR(25),TDI(10),TN(11),TD(11),HN(10),
     1 HD(5),TNHD(11),TNHN(11),TDHD(11),RTRL(25),RTIM(25),BAR(25),
     2 COLE(25,25),VAR(25),CN(10),CD(10),GN(25)
      DBTOF(BSP)=EXPF(0.115129*BSP)
      TODBF(PSB)=8.6858*LOGF(PSB)
C     INPUT DATA SECTION READS INPUT INFORMATION AND SPECIFICATIONS,
C     AND CHECKS THE VALIDITY OF REQUIREMENTS AGAINST THE LIMITATIONS
C     0F THE PROGRAM. ALSO COMPUTES THE UNCOMPENSATED CLOSED LOOP TRANS-
C     FER FUNCTION AND THE COEFFICIENTS OF THE STEADY STATE ERROR
C     EQUATION. ERROR AND INFORMATION PRINTOUTS ARE SELF-EXPLANATORY.
      PRINT 3
    3 FORMAT (1H1,////,20X,11H INPUT DATA    )
      PRINT 4
    4 FORMAT (19X,13H ************   )
C     THIS SECTION READS THE NUMERATOR AND DENOMINATOR OF THE OPEN LOOP
C     TRANSFER FUNCTION, EITHER AS THE ROOTS OR THE COEFFICIENTS OF
C     POLYNOMIALS.
      PRINT 5
```

```
   5 FORMAT (1H0,///,15X,35H OPEN LOOP TRANSFER FUNCTION - T(S)   )
     READ 10, ICORT,NNT,NDT,KRHP
  10 FORMAT (4I5)
     MARK=0
     KTNO=1
     KTDO=1
     KTD1=1
     NNTI=NNT-ICORT
     IF (NNTI-9) 12,12,11
  11 MARK=1
  12 NDTI=NDT-ICORT
     IF (NDTI-10) 13,13,14
  13 IF (NDTI-1) 14,14,1450
  14 MARK=1
1450 IF (MARK) 17,17,1500
1500 PRINT 1510, NNTI
1510 FORMAT (1H0,//,10X,24H ORDER OF NUMERATOR IS (,I2,3H).    )
     PRINT 1520, NDTI
1520 FORMAT (1H0,//,10X,26H ORDER OF DENOMINATOR IS (,I2,3H).    )
     IF (NDTI-1) 1530,1530,1580
1530 PRINT 1540
1540 FORMAT (1H0,//,6X,34H ERROR - ORDER OF DENOMINATOR MUST    )
     PRINT 1550
1550 FORMAT (1H0,13X,20H BE (2), OR GREATER.    )
     GO TO 9999
1580 NNTJ=(NNT+ICORT)/(ICORT+1)
     READ 1600, ((RTRL(I),RTIM(I)),I=1,NNTJ)
1600 FORMAT (8E10.4)
     NDTJ=(NDT+ICORT)/(ICORT+1)
     READ 1610, ((BAR(I),VAR(I)),I=1,NDTJ)
1610 FORMAT (8E10.4)
     GO TO 200
  17 PRINT 20
  20 FORMAT (1H0,//,10X,23H NUMERATOR OF OPEN LOOP    )
     PRINT 21
  21 FORMAT (1H0,9X,26H TRANSFER FUNCTION - TN(S)    )
```

```
      IF (ICORT) 30,30,140
   30 IF (NNTI) 32,32,35
   32 READ 34, TN(0)
   34 FORMAT (E10.4)
      TNO=TN(0)
      CALL DATAPRO (TNO,1)
      IF (TNO) 3410,3410,75
 3410 PRINT 3420
 3420 FORMAT (1H0,//,6X,41H ERROR - TN(S) READ AS ZERO OR A NEGATIVE    )
      PRINT 3430
 3430 FORMAT (1H0,13X,32H NUMBER. PROGRAM WILL NOT SOLVE.    )
      GO TO 9999
   35 READ 40, ((TNR(I),TNI(I)),I=1,NNTI)
   40 FORMAT (8E10.4)
      CALL DATAPR1 (NNTI,TNR,TNI,0)
      CALL RTARRG (NNTI,TNR,TNI,MARK)
      IF (MARK) 50,50,9999
   50 CALL RTEXPD (NNTI,TNR,TNI,TN)
      IF (TN(0)) 70,60,70
   60 KTNO=0
   70 CALL DATAPR2 (NNTI,TN,0,1)
      KOUT=0
      CALL RHPCHK (0,0,NNTI,TNR,KRHP,KOUT)
   75 READ 80, ((TDR(I),TDI(I)),I=1,NDTI)
   80 FORMAT (8E10.4)
      PRINT 90
   90 FORMAT (1H0,///,10X,25H DENOMINATOR OF OPEN LOOP    )
      PRINT 91
   91 FORMAT (1H0,9X,26H TRANSFER FUNCTION - TD(S)    )
      CALL DATAPR1 (NDTI,TDR,TDI,0)
      CALL RTARRG (NDTI,TDR,TDI,MARK)
      IF (MARK) 100,100,9999
  100 CALL RTEXPD (NDTI,TDR,TDI,TD)
      IF (TD(0)) 130,110,130
  110 KTDO=0
      IF (TD(1)) 130,120,130
```

```
120 KTD1=0
130 CALL DATAPR2 (NDTI,TD,0,2)
    CALL RHPCHK (0,1,NDTI,TDR,KRHP,KOUT)
    IF (KOUT-KRHP) 200,200,9999
140 IF (NNTI) 142,142,145
142 READ 144, TN(0)
144 FORMAT (E10.4)
    TNO=TN(0)
    CALL DATAPRO (TNO,1)
    IF (TNO) 3410,3410,160
145 READ 150, TN(0),(TN(I),I=1,NNTI)
150 FORMAT (8E10.4)
    CALL DATAPR2 (NNTI,TN,1,1)
    NOPR=-1
    KOUT=0
    IF (NNTI-1) 151,151,152
151 TNR(1)=-TN(0)
    GO TO 154
152 PRINT 153
153 FORMAT (1H0,//,10X,15H ROOTS OF TN(S)    )
    CALL ROOTFND (NNTI,TN,TNR,TNI,NOPR)
    CALL DATAPR1 (NNTI,TNR,TNI,1)
154 CALL RHPCHK (1,0,NNTI,TNR,KRHP,KOUT)
    IF (TN(0)) 160,155,160
155 KTNO=0
160 PRINT 170
170 FORMAT (1H0,///,10X,25H DENOMINATOR OF OPEN LOOP    )
    PRINT 171
171 FORMAT (1H0,9X,26H TRANSFER FUNCTION - TD(S)    )
175 READ 180, TD(0),(TD(I),I=1,NDTI)
180 FORMAT (8E10.4)
    CALL DATAPR2 (NDTI,TD,1,2)
    NOPR=-1
    PRINT 181
181 FORMAT (1H0,//,10X,15H ROOTS OF TD(S)    )
    CALL ROOTFND (NDTI,TD,TDR,TDI,NOPR)
```

```
      CALL DATAPR1 (NDTI,TDR,TDI,1)
      CALL RHPCHK (1,1,NDTI,TDR,KRHP,KOUT)
      IF (KOUT-KRHP) 184,184,9999
  184 IF (TD(0)) 200,185,200
  185 KTD0=0
      IF (TD(1)) 200,190,200
  190 KTD1=0
C     THIS SECTION READS THE ORDER OF FEEDBACK, AND THE COEFFICIENTS OF
C     THE FEEDBACK DEVICE USED.
  200 READ 210, LFBK,NNH,NDH
  210 FORMAT (3I5)
      PRINT 215
  215 FORMAT (1H0,/////,15X,39H FEEDBACK PATH TRANSFER FUNCTION - H(S) )
      PRINT 2151, LFBK
 2151 FORMAT (1H0,/,10X,2H (,I1,30H) ORDER FEEDBACK IS SPECIFIED.  ,/   )
      NNHM1=NNH-1
      NDHM1=NDH-1
      LFPNH=LFBK+NNHM1
      NTPH=NNTI+LFPNH
      MTPH=NDTI+NDHM1
      NTP1=NTPH+1
      IF (MARK) 229,229,216
  216 PRINT 217, LFPNH
  217 FORMAT (1H0,//,10X,24H ORDER OF NUMERATOR IS (,I2,3H).    )
      PRINT 218, NDHM1
  218 FORMAT (1H0,//,10X,26H ORDER OF DENOMINATOR IS (,I2,3H).    )
  220 IF (MTPH-NTP1) 221,224,224
  221 PRINT 222
  222 FORMAT(1H0,//,6X,43H ERROR - ORDER OF PRODUCT OF OPEN LOOP AND    )
      PRINT 2221
 2221 FORMAT (1H0,13X,38H FEEDBACK DENOMINATORS MUST BE GREATER    )
      PRINT 223
  223 FORMAT (1H0,13X,37H THAN ORDER OF PRODUCT OF NUMERATORS.    )
      PRINT 2231, NNTI,LFPNH,NTPH
 2231 FORMAT (1H0,/,14X,31H ORDER OF NUMERATOR PRODUCT IS ,I4,3H + ,I2,
     1 3H = ,I2    )
```

```
      PRINT 2232, NDTI,NDHM1,MTPH
 2232 FORMAT (1H0,/,14X,33H ORDER OF DENOMINATOR PRODUCT IS ,I2,3H + ,
     1 I2,3H = ,I2,// )
      MARK=1
  224 IF (10-MTPH) 225,228,228
  225 PRINT 226
  226 FORMAT(1H0,//,6X,43H ERROR - ORDER OF PRODUCT OF OPEN LOOP AND    )
      PRINT 2661
 2661 FORMAT (1H0,13X,35H FEEDBACK DENOMINATORS MUST BE LESS    )
      PRINT 227
  227 FORMAT (1H0,13X,36H THAN 11. ORDER OF PRODUCT FOR DATA     )
      PRINT 2271, NDTI,NDHM1,MTPH
 2271 FORMAT (1H0,13X,9H GIVEN IS ,I5,3H    ,I2,3H = ,I2    )
      MARK=1
  228 IF (MARK) 400,400,9999
  229 READ 230, HN(0),HN(1),HN(2)
  230 FORMAT (3E10.4)
      READ 240, HD(0),HD(1),HD(2)
  240 FORMAT (3E10.4)
      IF (LFBK) 280,280,260
  260 NNHP1=NNH+1
      NPLP1=NNHP1+LFBK
      DO 270 I=1,NNHP1
      J=NPLP1-I
      K=J-LFBK
      HN(J)=HN(K)
  270 HN(K)=0.0
  280 PRINT 290
  290 FORMAT (1H0,/,10X,27H NUMERATOR OF FEEDBACK PATH    )
      PRINT 291
  291 FORMAT (1H0,9X,26H TRANSFER FUNCTION - HN(S)    )
      IF (LFPNH) 300,300,310
  300 HN0=HN(0)
      CALL DATAPRO (HN0,3)
      GO TO 320
  310 CALL DATAPR2 (LFPNH,HN,1,3)
  320 PRINT 330
  330 FORMAT (1H0,//,10X,29H DENOMINATOR OF FEEDBACK PATH     )
      PRINT 331
  331 FORMAT (1H0,9X,26H TRANSFER FUNCTION - HD(S)    )
      IF (NDHM1) 340,340,350
  340 HD0=HD(0)
      CALL DATAPRO (HD0,4)
      GO TO 220
  350 CALL DATAPR2 (NDHM1,HD,1,4)
      GO TO 220
C     THIS SECTION COMPUTES AND PRINTS OUT THE UNCOMPENSATED CLOSED
C     LOOP TRANSFER FUNCTION.
  400 PRINT 405
  405 FORMAT (1H0,/////,15X,31H CLOSED LOOP TRANSFER FUNCTION    )
      PRINT 406
  406 FORMAT (1H0,18X,21H G(S) - UNCOMPENSATED   )
      PRINT 410
  410 FORMAT (1H0,//,10X,26H NUMERATOR OF CLOSED LOOP    )
      PRINT 411
  411 FORMAT (1H0,9X,27H TRANSFER FUNCTION - GN(S)    )
      CALL PLYMPY (NNTI,TN,NDHM1,HD,N1N,TNHD)
      IF (N1N) 420,420,430
  420 VBLE=TNHD(0)
      CALL DATAPR3 (0,VBLE,0)
      GO TO 445
  430 N1NP1=N1N+1
      DO 440 I=1,N1NP1
      J=N1NP1-I
      VBLE =TNHD(J)
      CALL DATAPR3 (J,VBLE,N1N)
  440 CONTINUE
  445 PRINT 450
  450 FORMAT (1H0,///,10X,28H DENOMINATOR OF CLOSED LOOP    )
      PRINT 451
  451 FORMAT (1H0,9X,27H TRANSFER FUNCTION - GD(S)    )
      CALL PLYMPY (NDTI,TD,NDHM1,HD,N1D,TDHD)
      CALL PLYMPY (NNTI,TN,LFPNH,HN,N2D,TNHN)
      N1MN2=N1D-N2D
      N1DP1=N1D+1
      DO 460 I=1,N1MN2
      J=N1DP1-I
      VBLE=TDHD(J)
      CALL DATAPR4 (J,VBLE,N1D)
  460 CONTINUE
      N2DP1=N2D+1
      DO 470 I=1,N2DP1
      J=N2DP1-I
      VBLE1=TDHD(J)
      VBLE2=TNHN(J)
      CALL DATAPR5 (J,VBLE1,VBLE2)
  470 CONTINUE
C     THIS SECTION READS THE STEADY STATE ERROR SPECIFICATION AND
C     COMPUTES THE COEFFICIENTS OF THE ERROR EQUATION.
  510 PRINT 520
  520 FORMAT (1H0,/////,15X,33H STEADY STATE ERROR SPECIFICATION   )
      READ 530, KLN,ERR
  530 FORMAT (I5,E10.4)
      IF (KLN) 531,531,575
  531 PRINT 532
  532 FORMAT (1H0,//,10X,35H NO STEADY STATE ERROR REQUIREMENT   )
      PRINT 533
  533 FORMAT (1H0,9X,15H IS SPECIFIED.   )
      ECDO=0.0
      EKXNO=-1.0
      EKNO=1.0
      GO TO 581
  575 PRINT 580, ERR
  580 FORMAT (1H0,/,10X,35H ERROR SPECIFIED (ERROR / INPUT) IS ,E11.4   )
      KLN=1
      CALL SSERR1 (KLN,ERR,LFBK,TNHD,TDHD,TNHN,KTNO,KTDO,KTD1,ECDO,
     1 EKXNO,EKNO,MARK)
      IF (MARK)  581,581,9999
```

```
C     THIS SECTION READS THE BANDWIDTH SPECIFICATION.
  581 PRINT 583
  583 FORMAT (1H0,/////,15X,24H BANDWIDTH SPECIFICATION   )
      READ 584, BDSPC,BOMEG
  584 FORMAT (2E10.4)
      IF (BOMEG) 585,585,595
  585 PRINT 586
  586 FORMAT (1H0,//,10X,39H NO BANDWIDTH REQUIREMENT IS SPECIFIED.    )
      MKBD=0
      MKLN=KLN+MKBD
      IF (MKLN) 587,587,600
  587 PRINT 588
  588 FORMAT (1H0,//,6X,48H NOTE - DATA INDICATES NO BANDWIDTH OR PARAME
     1TER   )
      PRINT 589
  589 FORMAT (1H0,12X,45H DEPENDENT STEADY STATE ERROR SPECIFICATIONS. )
      PRINT 590
  590 FORMAT (1H0,12X,44H UNDER THESE CONDITIONS THE PROGRAM WILL NOT  )
      PRINT 591
  591 FORMAT (1H0,12X,41H SOLVE FOR FORWARD GAIN AND FEEDBACK GAIN   )
      PRINT 592
  592 FORMAT (1H0,12X,43H SEPARATELY. SOLUTION IS FOR THE PRODUCT OF   )
      PRINT 593
  593 FORMAT (1H0,12X,20H THE GAINS.  (KT*KX)    )
      GO TO 600
  595 MKBD=2
      MKLN=KLN+MKBD
      PRINT 598, BDSPC
  598 FORMAT (1H0,//,10X,8H GAIN OF ,1PE11.3,17H DB  AT FREQUENCY   )
      PRINT 599, BOMEG
  599 FORMAT (1H0,9X,3H OF,1PE11.3,22H RAD/SEC IS SPECIFIED.   )
C     THIS SECTION READS THE CLOSED LOOP ROOTS SPECIFIED, AS ZETA AND
C     OMEGAN, AND/OR INDIVIDUAL ROOTS.
  600 READ 610,ISPEC,JRSPC
  610 FORMAT (2I5)
      PRINT 615
```

```
 615 FORMAT (1H0,/////,15X,20H ROOTS SPECIFIED FOR    )
     PRINT 6150
6150 FORMAT (1H0,14X,21H CLOSED LOOP SOLUTION    )
     I4=1
     IF (JRSPC) 616,616,620
 616 PRINT 617
 617 FORMAT (1H0,//,6X,41H ERROR - NO ROOTS ARE SPECIFIED. AT LEAST    )
     GO TO 736
 620 IF (ISPEC) 625,625,650
 625 READ 630, ZETA,OMEGAN
 630 FORMAT (2E10.4)
     PRINT 632, ZETA,OMEGAN
 632 FORMAT(1H0,//,10X,7H ZETA =,E11.4,6X,8H OMEGA =,1PE12.4    )
     RTRL(1)=(-1.0)*ZETA*OMEGAN
     RTRL(2)=RTRL(1)
     SZETA=SQRTF (1.0-(ZETA**2))
     RTIM(1)=SZETA*OMEGAN
     RTIM(2)=-RTIM(1)
     IF (JRSPC-2) 635,665,640
 635 JRSPC=2
     GO TO 665
 640 I4=3
 650 READ 660, ((RTRL(I),RTIM(I)),I=I4,JRSPC)
 660 FORMAT (8E10.4)
 665 CALL DATAPR1 (JRSPC,RTRL,RTIM,0)
     DO 670 I=1,JRSPC
     IF (RTRL(I)) 670,670,666
 666 PRINT 667
 667 FORMAT (1H0,//,6X,44H ERROR - ROOT(S) SPECIFIED IN THE RIGHT HALF)
     PRINT 668
 668 FORMAT (1H0,13X,31H PLANE. PROGRAM WILL NOT SOLVE.  )
     GO TO 9999
 670 CONTINUE
     CALL RTARRG (JRSPC,RTRL,RTIM,MARK)
     IF (MARK) 675,675,9999
C    SOLUTION SECTION COMPUTES THE SOLUTION AND PRINTS OUT THE RESULTS.
```

```
 675 PRINT 677
 677 FORMAT (1H1,////,20X,24H SOLUTION OF THE PROBLEM    )
     PRINT 678
 678 FORMAT (1H0,18X,26H *************************    )
     IMX=(2*N1D)-1
     IMLK=0
 680 PRINT 690,IMX
 690 FORMAT(1H0,///,11X,16HPROGRAM REQUIRES,I3,20H ROOTS FOR SOLUTION.)
     IF (IMX-JRSPC) 700,711,720
 700 PRINT 710, JRSPC
 710 FORMAT (1H0,//,6X,9H ERROR - ,I2,21H ROOTS ARE SPECIFIED,    )
     PRINT 7101, IMX
7101 FORMAT (1H0,13X,10H BUT ONLY ,I2,21H ROOTS ARE REQUIRED.    )
     GO TO 9999
 711 IF (MKLN-2) 720,720,712
 712 PRINT 713
 713 FORMAT (1H0,//,6X,47H NOTE - SINCE ALL REQUIRED ROOTS ARE SPECIFIE
    1D,    )
     PRINT 714
 714 FORMAT (1H0,12X,38H BANDWIDTH FOR THIS PROBLEM IS A FIXED  )
     PRINT 715
 715 FORMAT (1H0,12X,38H CONSTANT. BANDWIDTH SPECIFICATION MAY  )
     PRINT 716
 716 FORMAT (1H0,12X,39H NOT BE SATISFIED. CHECK OUTPUT BELOW.     )
     IMLK=4
 720 BASE=RTRL(1)
     DO 725 I=2,JRSPC
     IF (RTRL(I)-BASE) 722,725,725
 722 BASE=RTRL(I)
 725 CONTINUE
     FAQ=ABSF(RTIM(1))
     DO 730 I=2,JRSPC
     FAB=ABSF(RTIM(I))
     IF (FAB-FAQ) 727,728,728
 727 IF (FAB) 730,730,729
 728 IF (FAQ) 729,729,730
 729 FAQ=FAB
 730 CONTINUE
     IF (FAQ) 733,733,740
 733 IF (BASE) 738,735,666
 735 PRINT 7350
7350 FORMAT (1H0,//,6X,43H ERROR - ROOTS SPECIFIED ARE ZERO. AT LEAST )
 736 PRINT 7360
7360 FORMAT (1H0,13X,37H ONE NON-ZERO ROOT MUST BE SPECIFIED    )
     PRINT 7370
7370 FORMAT (1H0,13X,34H TO ESTABLISH DOMINANCE CRITERION.    )
     GO TO 9999
 738 FAQ=-BASE
     GO TO 745
 740 IF (BASE) 745,742,666
 742 BASE=-FAQ
 745 FAQ=0.1*FAQ
     FCK=3.0
     FCJ=1.0
 750 IF (MKLN-2) 917,751,751
C     THIS SECTION OBTAINS AND PRINTS OUT THE SOLUTION FOR BANDWIDTH,
C     BY ITERATION IF NECESSARY.
 751 PRINT 752
 752 FORMAT (1H0,/////,15X,19H BANDWIDTH SOLUTION    )
     IF (MKLN-2) 917,755,760
 755 IRGEN=IMX-JRSPC
     IF (IRGEN-1) 7555,7552,7555
7552 FCK=FCK-1.0
7555 FCJ=1.0
     CALL BANDSLV (N1N,TNHD,NID,TDHD,N2D,TNHN,ECDO,EKXNO,EKNO,JRSPC,
    1 FCK,FCJ,BASE,FAQ,RTRL,RTIM,BOMEG,GMGN,GMGD,GMGT,MARK)
     IF (MARK) 757,757,9999
 757 GMDB=BDSPC
     RLER=0.0
     MCML=0
     GO TO 900
 760 IF (IMLK) 770,770,765
 765 FCJ=0.0
     CALL BANDSLV (N1N,TNHD,N1D,TDHD,N2D,TNHN,ECDO,EKXNO,EKNO,JRSPC,
    1 FCK,FCJ,BASE,FAQ,RTRL,RTIM,BOMEG,GMGN,GMGD,GMGT,MARK)
     IF (MARK) 766,766,9999
 766 GMDB=TODBF (GMGT)
7660 IF (BDSPC) 7662,7661,7662
7661 RLER=GMDB
     ABRL=100.0*ABSF(RLER)
     GO TO 7663
7662 GMPC=GMDB-BDSPC
     RLER=(100.0*GMPC)/ABSF(BDSPC)
     ABRL=ABSF(RLER)
7663 MCML=0
     IF (1.0-ABRL) 767,900,900
 767 MCML=5
     GO TO 900
 770 IF (BDSPC) 7701,7702,7704
7701 BDUL=0.99*BDSPC
     BDLL=1.01*BDSPC
     GO TO 7706
7702 BDUL=0.01
     BDLL=-0.01
     GO TO 7706
7704 BDUL=1.01*BDSPC
     BDLL=0.99*BDSPC
7706 BDZ=DBTOF (BDUL)
     BDA=DBTOF (BDLL)
     FCJ=0.0
     LTRY=500
     MCLIM=LTRY
     MCT=1
     NCT=0
     MCML=0
     AP=-1.0
     BLFJ=1.0
     CALL BANDSLV (N1N,TNHD,N1D,TDHD,N2D,TNHN,ECDO,EKXNO,EKNO,JRSPC,
   1 FCK,FCJ,BASE,FAQ,RTRL,RTIM,BOMEG,GMGN,GMGD,GMGT,MARK)
     IF (MARK) 773,773,9999
 773 IF (BDZ-GMGT) 785,766,775
 775 IF (GMGT-BDA) 820,766,766
 780 LTRY=11
     GMDB=TODBF (GMGT)
     CALL ITERPR (NCT,MCT,GMDB,BDSPC,RLER)
     NCT=10
 785 AP=-AP
     DLFJ=DLFJ/10.0
     IF (MCT-MCLIM) 787,875,875
 787 DO 800 I=1,LTRY
     FCJ=FCJ+(AP*DLFJ)
     CALL BANDSLV (N1N,TNHD,N1D,TDHD,N2D,TNHN,ECD0,EKXN0,EKN0,JRSPC,
   1 FCK,FCJ,BASE,FAQ,RTRL,RTIM,BOMEG,GMGN,GMGD,GMGT,MARK)
     IF (BDZ-GMGT) 795,855,790
 790 IF (GMGT-BDA) 810,855,855
 795 MCT=MCT+1
 800 CONTINUE
     GO TO 875
 810 LTRY=11
     GMDB=TODBF (GMGT)
     CALL ITERPR (NCT,MCT,GMDB,BDSPC,RLER)
     NCT=10
 820 AP=-AP
     DLFJ=DLFJ/10.0
     IF (MCT-MCLIM) 840,875,875
 840 DO 870 I=1,LTRY
     FCJ=FCJ+(AP*DLFJ)
     CALL BANDSLV (N1N,TNHD,N1D,TDHD,N2D,TNHN,ECD0,EKXN0,EKN0,JRSPC,
   1 FCK,FCJ,BASE,FAQ,RTRL,RTIM,BOMEG,GMGN,GMGD,GMGT,MARK)
     IF (BDA-GMGT) 850,855,860
 850 IF (BDZ-GMGT) 780,855,855
 855 GMDB=TODBF(GMGT)
     CALL ITERPR (NCT,MCT,GMDB,BDSPC,RLER)
     GO TO 7660
```

```
860 MCT=MCT+1
870 CONTINUE
875 MCML=5
    IF (NCT) 880,880,895
880 GMDB=TODBF(GMGT)
    NCT=-10
    CALL ITERPR (NCT,MCT,GMDB,BDSPC,RLER)
    PRINT 890
890 FORMAT (1H0,//,7X,44HNOTE - ITERATIVE SOLUTION DOES NOT CONVERGE.)
    PRINT 891
891 FORMAT (1H0,12X,37H DOMINANCE OF THE SPECIFIED ROOTS CAN    )
    PRINT 892
892 FORMAT (1H0,12X,34H NOT BE ASSURED WITH THE SPECIFIED    )
    PRINT 893
893 FORMAT (1H0,12X,21H BANDWIDTH CRITERION.    )
    GO TO 9999
895 PRINT 896
896 FORMAT (1H0,//,6X,42H NOTE - PROCESS HAS FAILED TO SOLVE WITHIN  )
    PRINT 897
897 FORMAT (1H0,12X,36H ACCEPTED ERROR LIMITS (1.0 PERCENT)    )
    PRINT 898,MCLIM
898 FORMAT (1H0,12X,17H IN THE ALLOTED (I4,14H ) ITERATIONS.    )
900 PRINT 905, GMDB,BOMEG
905 FORMAT (1H0,//,10X,9H GAIN IS ,1PE10.3,17H DB  AT FREQUENCY    ,
   1 1PE10.3,8H RAD/SEC    )
    PRINT 906, RLER
906 FORMAT (1H0,/,10X,18H RELATIVE ERROR IS ,1PE11.3,8H PERCENT    )
    IF (MCML) 909,909,911
909 PRINT 910
910 FORMAT (1H0,/,10X,38H BANDWIDTH SPECIFICATION IS SATISFIED.    )
    GO TO 917
911 PRINT 912
912 FORMAT(1H0,/,10X,42H BANDWIDTH SPECIFICATION IS NOT SATISFIED.    )
917 CALL ROOTGEN (JRSPC,IMX,FCK,FCJ,BASE,FAQ,RTRL,RTIM)
920 CALL RTEXPD (IMX,RTRL,RTIM,BAR)
    CALL BARARRG (IMX,BAR)
```

```
      CALL COPLUG (N1D,TDHD,N2D,TNHN,NMAT,COLE)
      LX1=NMAT+1
      LX2=N1D+1
      COLE(LX1,1)=ECDO
      COLE(LX1,LX2)=EKXNO
      COLE(LX1,LX1)=EKNO
      CALL LINSOLV (LX1,COLE,BAR,VAR,MARK)
      IF (MARK) 925,925,9999
C     THIS SECTION CHECKS AND PRINTS OUT THE SOLUTION FOR STEADY STATE
C     ERROR.
  925 IF (KLN) 950,950,935
  935 TERRD=(TDHD(0)*VAR(1))+(TNHN(0)*VAR(LX2))
      TERRN=TERRD-(TNHD(0)*VAR(LX1))
      TERR=TERRN/TERRD
      PRINT 9391
 9391 FORMAT (1H0,/////,15X,28H STEADY STATE ERROR SOLUTION    )
  940 PRINT 941,TERR
  941 FORMAT (1H0,//,10X,23H STEADY STATE ERROR IS ,E10.4 )
      IF (ERR) 9412,9411,9412
 9411 IF (TERR) 946,944,946
 9412 RLER=(100.0*(TERR-ERR))/ERR
      PRINT 942
  942 FORMAT (1H0,/,10X,23H RELATIVE SPECIFICATION    )
      PRINT 943, RLER
  943 FORMAT (1H0,9X,9H ERROR IS ,1PE11.3,8H PERCENT    )
      ABREL=ABSF (RLER)
      IF (ABREL-1.0) 944,944,946
  944 PRINT 945
  945 FORMAT (1H0,/,10X,34H ERROR SPECIFICATION IS SATISFIED.    )
      GO TO 950
  946 PRINT 947
  947 FORMAT (1H0,/,10X,38H ERROR SPECIFICATION IS NOT SATISFIED.    )
C     THIS SECTION CHECKS AND INDICATES THE ACCURACY OF THE SOLUTION
C     OF THE SYSTEM OF LINEAR EQUATIONS.
  950 PRINT 951
  951 FORMAT (1H0,/////,15X,43H SOLUTION OF THE SYSTEM OF LINEAR EQUATIO
   1NS    )
      CALL RTEXPD (IMX,RTRL,RTIM,BAR)
      CALL BARARRG (IMX,BAR)
      CALL COPLUG (N1D,TDHD,N2D,TNHN,NMAT,COLE)
      PRINT 952
  952 FORMAT (1H0,//,10X,9H VARIABLE ,9X,11H PARAMETERS ,6X,15H RELATIVE
     1 ERROR    )
      DO 965 I=1,NMAT
      IM1=I-1
      CLX=0.0
      DO 955 J=1,NMAT
  955 CLX=CLX+(COLE(I,J)*VAR(J))
      IF (BAR(I)) 957,956,957
  956 RELCER=CLX
      GO TO 958
  957 CER=CLX-BAR(I)
      RELCER=(100.0*CER)/BAR(I)
  958 IF (I-N1D) 960,960,963
  960 PRINT 961, I,IM1,RELCER
  961 FORMAT (1H0,/,10X,11H VARIABLE (,I2,2H) ,5X,4H CD(,I1,2H) ,11X,
     1 1PE10.3, 8H PERCENT   )
      GO TO 965
  963 IMJ=IM1-N1D
      PRINT 964,I,IMJ,RELCER
  964 FORMAT (1H0,/,10X,11H VARIABLE (,I2,2H) ,5X,10H KT*KX*CN(,I1,2H) ,
     1 5X,1PE10.3, 8H PERCENT    )
  965 CONTINUE
      IF (KLN) 9660,9660,9650
 9650 RELCER=RLER
      PRINT 966, LX1,MARK,RELCER
  966 FORMAT (1H0,/,10X,11H VARIABLE (,I2,2H) ,5X,7H KT*CN(,I2,2H) ,7X,
     1 1PE10.3,8H PERCENT    )
 9660 PRINT 9670
 9670 FORMAT (1H0,//,6X,35H NOTE - RELATIVE ERROR STATED ABOVE    )
      PRINT 9680
 9680 FORMAT (1H0,12X,29H MEASURES THE ACCURACY OF THE    )
```

```
      PRINT 9690
 9690 FORMAT (1H0,12X,28H SOLUTION FOR EACH VARIABLE.   )
C     THIS SECTION COMPUTES AND PRINTS OUT THE GAINS AND THE
C     COEFFICIENTS AND ROOTS OF THE FILTER.
      PRINT 968
  968 FORMAT (1H0,/////,15X,35H SOLUTION FOR INDIVIDUAL PARAMETERS    )
      CALL PARSOLV (N1D,VAR,CN,CD,GKX,GKT)
      PRINT 970
  970 FORMAT (1H0,//,10X,34H NUMERATOR OF CASCADE COMPENSATION    )
      N1DM1=N1D-1
      CALL DATAPR2 (N1DM1,CN,1,5)
      IF (N1DM1-2) 974,972,972
  972 PRINT 973
  973 FORMAT (1H0,//,10X,20H ZEROS OF THE FILTER    )
      CALL ROOTFND (N1DM1,CN,TDR,TDI,0)
  974 PRINT 975
  975 FORMAT (1H0,///,10X,36H DENOMINATOR OF CASCADE COMPENSATION    )
      CALL DATAPR2 (N1DM1,CD,1,6)
      IF (N1DM1-2) 9800,977,977
  977 PRINT 978
  978 FORMAT (1H0,//,10X,20H POLES OF THE FILTER    )
      CALL ROOTFND (N1DM1,CD,TDR,TDI,1)
 9800 IF (MKLN-2) 985,980,990
  980 GKTKX=GKT
      GMGX= DBTOF (BDSPC)
      GKT= (GMGX*GMGD)/GMGN
      GKX=GKTKX/GKT
      GO TO 990
  985 IF (MKLN) 986,986,990
  986 PRINT 987, GKT
  987 FORMAT (1H0,//,10X,27H PRODUCT OF GAINS (KT*KX) = ,1PE11.4    )
      PRINT 988, GKT
  988 FORMAT (1H0,/,10X,22H FORWARD GAIN (KT) IS  ,1PE11.4,2H,    )
      PRINT 989
  989 FORMAT (1H0,9X,36H WITH UNITY FEEDBACK GAIN.  (KX=1.0)    )
      GO TO 995
  990 PRINT 991,GKT
  991 FORMAT (1H0,//,10X,20H FORWARD GAIN (KT) = ,1PE11.4    )
      PRINT 992,GKX
  992 FORMAT (1H0,/,10X,21H FEEDBACK GAIN (KX) = ,1PE11.4    )
C     THIS SECTION COMPUTES AND PRINTS OUT THE COEFFICIENTS AND ROOTS
C     OF THE CLOSED LOOP TRANSFER FUNCTION.
  995 PRINT 996
  996 FORMAT (1H0,/////,15X,30H CLOSED LOOP TRANSFER FUNCTION    )
      PRINT 997
  997 FORMAT (1H0,21X,18H OF SOLVED PROBLEM    )
      PRINT 998
  998 FORMAT (1H0,//,10X,21H NUMERATOR POLYNOMIAL    )
      CALL PLYMPY (N1N,TNHD,N1DM1,CN,N4D,GN)
      N4DP1=N4D+1
      DO 999 I=1,N4DP1
      J=I 1
  999 GN(J)=GKT*GN(J)
      CALL DATAPR2 (N4D,GN,1,7)
      IF (N4D) 1005,1005,1000
 1000 PRINT 1002
 1002 FORMAT (1H0,//,10X,31H ZEROS OF THE TRANSFER FUNCTION    )
      DO 1004 I=1,N4DP1
      J=I-1
 1004 GN(J)=GN(J)/GKT
      CALL ROOTFND (N4D,GN,TDR,VAR,1)
 1005 PRINT 1006
 1006 FORMAT (1H0,///,10X,23H DENOMINATOR POLYNOMIAL    )
      CALL REVARRG (IMX,BAR)
      CALL DATAPR2 (IMX,BAR,1,8)
      PRINT 1010
 1010 FORMAT (1H0,//,10X,30H POLES OF THE SYSTEM AND ROOTS    )
      PRINT 1011
 1011 FORMAT (1H0,9X,31H OF THE CHARACTERISTIC EQUATION    )
      CALL DATAPR1 (IMX,RTRL,RTIM,1)
 9999 END
C     SUBROUTINES DATAPR(0 TO 5) ARE PRINTOUT FORMATS FOR ROOTS AND
```

```
C     COEFFICIENTS OF POLYNOMIALS, AND FOR THE UNCOMPENSATED CLOSED
C     LOOP TRANSFER FUNCTION.
      SUBROUTINE DATAPRO (DNF,IXY)
   15 IF (IXY-2) 20,30,40
   20 PRINT 25,DNF
   25 FORMAT (1H0,/,10X,7H TN(S)= ,1PE12.4  )
      GO TO 100
   30 PRINT 35,DNF
   35 FORMAT (1H0,/,10X,7H TD(S)= ,1PE12.4  )
      GO TO 100
   40 IF (IXY-4) 45,55,65
   45 PRINT 50,DNF
   50 FORMAT (1H0,/,10X,7H HN(S)= ,1PE12.4  )
      GO TO 100
   55 PRINT 60,DNF
   60 FORMAT (1H0,/,10X,7H HD(S)= ,1PE12.4  )
      GO TO 100
   65 IF (IXY-6) 70,80,90
   70 PRINT 75,DNF
   75 FORMAT (1H0,/,10X,7H CN(S)= ,1PE12.4  )
      GO TO 100
   80 PRINT 85,DNF
   85 FORMAT (1H0,/,10X,7H CD(S)= ,1PE12.4  )
      GO TO 100
   90 PRINT 95,DNF
   95 FORMAT (1H0,/,10X,7H GN(S)= ,1PE12.4  )
  100 RETURN
      END
      SUBROUTINE DATAPR1 (NAB,AZR,AZI,KCORT)
      DIMENSION AZR(25),AZI(25),NC(25)
      IF (KCORT) 5,5,15
    5 PRINT 10,NAB
   10 FORMAT (1H0,/,10X,18H NUMBER OF ROOTS = ,I2)
   15 DO 20 I=1,NAB
   20 NC(I)=I
      I1=1
      MNAB=NAB
      IF (NAB-10) 29,29,23
   23 MNAB=10
   29 PRINT 30, (NC(I),I=I1,MNAB)
   30 FORMAT (1H0,/,2X,9H ROOT NO. ,I8,9I10 )
      PRINT 40, (AZR(I),I=I1,MNAB)
   40 FORMAT (1H0,/,2X,9H REAL PT. ,10E1 .2 )
      PRINT 50, (AZI(I),I=I1,MNAB)
   50 FORMAT (1H0,/,2X,9H IMAG PT. ,10E1 .2,/)
      IF (NAB-MNAB) 60,60,55
   55 MNAB=NAB
      I1=11
      GO TO 29
   60 RETURN
      END
      SUBROUTINE DATAPR2 (NAB,AZ,JCORT,IXY)
      DIMENSION AZ(25)
      IF (JCORT) 5,5,15
    5 PRINT 10
   10 FORMAT (1H0,//,10X,36H COEFFICIENTS OF EXPANDED POLYNOMIAL    )
   15 IF (IXY-2) 20,30,40
   20 PRINT 25, AZ(NAB),NAB
   25 FORMAT (1H0,/,10X,7H TN(S)= ,1PE12.4,3H S(,I2,2H)    )
      GO TO 115
   30 PRINT 35, AZ(NAB),NAB
   35 FORMAT (1H0,/,10X,7H TD(S)= ,1PE12.4,3H S(,I2,2H)    )
      GO TO 115
   40 IF (IXY-4) 45,55,65
   45 PRINT 50, AZ(NAB),NAB
   50 FORMAT (1H0,/,10X,7H HN(S)= ,1PE12.4,3H S(,I2,2H)    )
      GO TO 115
   55 PRINT 60, AZ(NAB),NAB
   60 FORMAT (1H0,/,10X,7H HD(S)= ,1PE12.4,3H S(,I2,2H)    )
      GO TO 115
   65 IF (IXY-6) 70,80,90
   70 PRINT 75, AZ(NAB),NAB
```

```
 75 FORMAT (1H0,/,10X,7H CN(S)= ,1PE12.4,3H S(,I2,2H)    )
    GO TO 115
 80 PRINT 85, AZ(NAB),NAB
 85 FORMAT (1H0,/,10X,7H CD(S)= ,1PE12.4,3H S(,I2,2H)    )
    GO TO 115
 90 IF (IXY-8) 95,105,105
 95 PRINT 100, AZ(NAB),NAB
100 FORMAT (1H0,/,10X,7H GN(S)= ,1PE12.4,3H S(,I2,2H)    )
    GO TO 115
105 PRINT 110, AZ(NAB),NAB
110 FORMAT (1H0,/,10X,7H GD(S)= ,1PE12.4,3H S(,I2,2H)    )
115 IF (NAB-1) 150,150,125
125 NABM1=NAB-1
    DO 140 I=1,NABM1
    J=NAB-I
    PRINT 130, AZ(J),J
130 FORMAT (1H0,/,16X,2H +,1PE11.4,3H S(,I2,2H)    )
140 CONTINUE
150 PRINT 160, AZ(0)
160 FORMAT (1H0,/,16X,2H +,1PE11.4    )
    RETURN
    END
    SUBROUTINE DATAPR3 (IJK,VBLE,KGE)
    IF (KGE-IJK) 10,10,60
 10 IF (IJK) 20,20,40
 20 PRINT 30,VBLE
 30 FORMAT (1H0,/,10X,12H GN(S)= KT*  ,1PE11.4  )
    GO TO 110
 40 PRINT 50, VBLE,IJK
 50 FORMAT (1H0,/,10X,12H GN(S)= KT*( ,1PE12.4,3H S(,I2,2H)    )
    GO TO 110
 60 IF (IJK) 70,70,90
 70 PRINT 80, VBLE
 80 FORMAT (1H0,/,21X,2H + ,1PE11.4,3H )     )
    GO TO 110
 90 PRINT 100, VBLE,IJK
```

```
100 FORMAT (1H0,/,21X,2H + ,1PE11.4,3H S( ,I2,2H)    )
110 RETURN
    END
    SUBROUTINE DATAPR4 (IJK,VBLE,KGE)
    IF (KGE-IJK) 10,10,30
 10 PRINT 20, VBLE,IJK
 20 FORMAT (1H0,/,10X,8H GD(S)= ,1PE11.4,3H S( ,I2,2H)    )
    GO TO 50
 30 PRINT 40, VBLE,IJK
 40 FORMAT (1H0,/,16X,2H + ,1PE11.4,3H S(,I2,2H)    )
 50 RETURN
    END
    SUBROUTINE DATAPR5 (IJK,VBLE1,VBLE2)
    IF (IJK) 10,10,30
 10 PRINT 20, VBLE1,VBLE2
 20 FORMAT (1H0,/,16X,4H + (,1PE11.4,9H + KT*KX*,1PE11.4,2H)    )
    GO TO 50
 30 PRINT 40,VBLE1,VBLE2,IJK
 40 FORMAT(1H0,/,17X,3H+ (,1PE11.4,9H   KT*KX*,1PE11.4,4H) S(,I2,2H) )
 50 RETURN
    END
    SUBROUTINE RTARRG (NROOTS,FACRL,FACIM,MARK)
C   RTARRG GROUPS ROOTS BY THEIR IMAGINARY PARTS, AND VERIFIES THAT
C   ALL COMPLEX ROOTS HAVE CONJUGATES.
    DIMENSION FACRL(25),FACIM(25)
    NRTS1=NROOTS/2
    NRTS2=2*NRTS1
    IF (NROOTS-NRTS2) 7,7,5
  5 NCORR=2
    GO TO 10
  7 NCORR=1
 10 LIMNR=NROOTS-NCORR
    DO 110 I=1,LIMNR,2
    K=I
    IP1=I+1
    BFIM=(-1.0)*ABSF(FACIM(I))
```

```
      DO 30 J=IP1,NROOTS
      FIMJ=(-1.0)*ABSF(FACIM(J))
      IF (FIMJ-BFIM) 20,30,30
   20 K=J
      BFIM=FIMJ
   30 CONTINUE
      IF (BFIM) 40,120,120
   40 DFIM=FACIM(K)
      AFRL=FACRL(K)
      AXRL=ABSF(AFRL*1.0E-06)
      DXIM=ABSF(DFIM*1.0E-06)
      HOLRL=FACRL(I)
      HOLIM=FACIM(I)
      FACRL(I)=FACRL(K)
      FACIM(I)=FACIM(K)
      FACRL(K)=HOLRL
      FACIM(K)=HOLIM
      NCNT=NROOTS-I
      NCHK=0
      DO 80 J=IP1,NROOTS
      CXIM=ABSF(FACIM(J)+DFIM)
      IF (CXIM-DXIM) 50,50,70
   50 CXRL=ABSF(FACRL(J)-AFRL)
      IF (CXRL-AXRL) 60,60,70
   60 L=J
      GO TO 100
   70 NCHK=NCHK+1
   80 CONTINUE
      IF (NCNT-NCHK) 90,90,100
   90 PRINT 95
   95 FORMAT (1H0,//,6X,44H ERROR - PROGRAM DOES NOT READ COMPLEX ROOTS)
      PRINT 96
   96 FORMAT (1H0,13X,33H AS CONJUGATES. CHECK DATA CARDS.  )
      MARK=1
      GO TO 120
  100 MARK=0
```

```
      HOLRL=FACRL(IP1)
      HOLIM=FACIM(IP1)
      FACRL(IP1)=FACRL(L)
      FACIM(IP1)=FACIM(L)
      FACRL(L)=HOLRL
      FACIM(L)=HOLIM
  110 CONTINUE
  120 RETURN
      END
      SUBROUTINE RTEXPD (NROOTS,FACRL,FACIM,SPCO)
C     RTEXPD MULTIPLIES THE ROOTS GIVEN TO FORM THE COEFFICIENTS OF A
C     POLYNOMIAL. SUBROUTINES TOTSUM, SUMPR(0 TO 12),AND CPXMPY ARE
C     PART OF RTEXPD.
      DIMENSION FACRL(25),FACIM(25),ACRL(13),ACIM(13),SP1(14),SP2(14),
     1 SPCO(25)
      MAX=NROOTS
      IF (MAX-12) 10,10,30
   10 DO 20 I=1,MAX
      CALL TOTSUM (MAX,I,FACRL,FACIM,SPFAC)
      J=MAX-I
   20 SPCO(J)=SPFAC
      SPCO(MAX)=1.0
      GO TO 80
   30 MAXM10=MAX-12
      DO 35 I=1,13
      ACRL(I)=0.0
   35 ACIM(I)=0.0
      DO 40 J=1,MAXM10
      JP=J+12
      ACRL(J)=FACRL(JP)
      FACRL(JP)=0.0
      ACIM(J)=FACIM(JP)
   40 FACIM(JP)=0.0
      DO 50 I=1,12
      CALL TOTSUM (12,I,FACRL,FACIM,SPFAC)
      J=12-I
```

```
 50 SP1(J)=SPFAC
    SP1(12)=1.0
    DO 60 I=1,MAXM10
    CALL TOTSUM (MAXM10,I,ACRL,ACIM,SPFAC)
    J=MAXM10-I
 60 SP2(J)=SPFAC
    SP2(MAXM10)=1.0
    CALL PLYMPY (12,SP1,MAXM10,SP2,MON,SPCO)
    MON=0
    DO 70 J=1,MAXM10
    JP=J+12
    FACRL(JP)=ACRL(J)
 70 FACIM(JP)=ACIM(J)
 80 RETURN
    END
    SUBROUTINE TOTSUM (MAX,I,FACRL,FACIM,SPFAC)
    DIMENSION NC(25),FACRL(25),FACIM(25)
    SUMRL=0.0
    SUMIM=0.0
280 IF (I-13) 290,285,285
285 L12=1
    CALL SUMP12 (L12,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
    GO TO 110
290 IF (I-12) 300,295,295
295 L11=1
    CALL SUMP11 (L11,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
    GO TO 110
300 IF (I-11) 310,305,305
305 L10=1
    CALL SUMP10 (L10,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
    GO TO 110
310 IF (I-10) 20,10,10
 10 L9=1
    CALL SUMPR9 (L9,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
    GO TO 110
 20 IF (I-9) 30,25,25
 25 L8=1
    CALL SUMPR8 (L8,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
    GO TO 110
 30 IF (I-8) 40,35,35
 35 L7=1
    CALL SUMPR7 (L7,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
    GO TO 110
 40 IF (I-7) 50,45,45
 45 L6=1
    CALL SUMPR6 (L6,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
    GO TO 110
 50 IF (I-6) 60,55,55
 55 L5=1
    CALL SUMPR5 (L5,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
    GO TO 110
 60 IF (I-5) 70,65,65
 65 L4=1
    CALL SUMPR4 (L4,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
    GO TO 110
 70 IF (I-4) 80,75,75
 75 L3=1
    CALL SUMPR3 (L3,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
    GO TO 110
 80 IF (I-3) 90,85,85
 85 L2=1
    CALL SUMPR2 (L2,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
    GO TO 110
 90 IF (I-2) 100,95,95
 95 L1=1
    CALL SUMPR1 (L1,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
    GO TO 110
100 L0=1
    CALL SUMPR0 (L0,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
110 SPFAC=SUMRL
    RETURN
    END
```

```
      SUBROUTINE SUMPR0 (L0,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
      DIMENSION NC(25),FACRL(25),FACIM(25)
      DO 10 N0=L0,MAX
      NC(1)=N0
      IP1=I+1
      DO 20 N50=IP1,25
   20 NC(N50)=0
      CALL CPXMPY (I,NC,FACRL,FACIM,PDRL,PDIM)
      SUMRL=SUMRL+PDRL
   10 SUMIM=SUMIM+PDIM
      RETURN
      END
      SUBROUTINE SUMPR1 (L1,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
      DIMENSION NC(25),FACRL(25),FACIM(25)
      LX1=MAX-1
      DO 10 N1=L1,LX1
      NC(2)=N1
      L0=N1+1
      CALL SUMPR0 (L0,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE SUMPR2 (L2,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
      DIMENSION NC(25),FACRL(25),FACIM(25)
      LX2=MAX-2
      DO 10 N2=L2,LX2
      NC(3)=N2
      L1=N2+1
      CALL SUMPR1 (L1,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE SUMPR3 (L3,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
      DIMENSION NC(25),FACRL(25),FACIM(25)
      LX3=MAX-3
      DO 10 N3=L3,LX3
      NC(4)=N3
      L2=N3+1
      CALL SUMPR2 (L2,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE SUMPR4 (L4,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
      DIMENSION NC(25),FACRL(25),FACIM(25)
      LX4=MAX-4
      DO 10 N4=L4,LX4
      NC(5)=N4
      L3=N4+1
      CALL SUMPR3 (L3,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE SUMPR5 (L5,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
      DIMENSION NC(25),FACRL(25),FACIM(25)
      LX5=MAX-5
      DO 10 N5=L5,LX5
      NC(6)=N5
      L4=N5+1
      CALL SUMPR4 (L4,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE SUMPR6 (L6,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
      DIMENSION NC(25),FACRL(25),FACIM(25)
      LX6=MAX-6
      DO 10 N6=L6,LX6
      NC(7)=N6
      L5=N6+1
      CALL SUMPR5 (L5,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE SUMPR7 (L7,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
      DIMENSION NC(25),FACRL(25),FACIM(25)
      LX7=MAX-7
      DO 10 N7=L7,LX7
      NC(8)=N7
      L6=N7+1
      CALL SUMPR6 (L6,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE SUMPR8 (L8,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
      DIMENSION NC(25),FACRL(25),FACIM(25)
      LX8=MAX-8
      DO 10 N8=L8,LX8
      NC(9)=N8
      L7=N8+1
      CALL SUMPR7 (L7,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE SUMPR9 (L9,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
      DIMENSION NC(25),FACRL(25),FACIM(25)
      LX9=MAX-9
      DO 10 N9=L9,LX9
      NC(10)=N9
      L8=N9+1
      CALL SUMPR8 (L8,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE SUMP10 (L10,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
      DIMENSION NC(25),FACRL(25),FACIM(25)
      LX1 =MAX-10
      DO 10 N10=L10,LX10
      NC(11)=N10
      L9=N10+1
      CALL SUMPR9 (L9,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE SUMP11 (L11,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
      DIMENSION NC(25),FACRL(25),FACIM(25)
      LX11=MAX-11
      DO 10 N11=L11,LX11
      NC(12)=N11
      L10=N11+1
      CALL SUMP10 (L10,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE SUMP12 (L12,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
      DIMENSION NC(25),FACRL(25),FACIM(25)
      LX12=MAX-12
      DO 10 N12=L12,LX12
      NC(13)=N12
      L11=N12+1
      CALL SUMP11 (L11,I,NC,MAX,FACRL,FACIM,SUMRL,SUMIM)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE CPXMPY (NMULT,NC,FACRL,FACIM,PDRL,PDIM)
      DIMENSION NC(25),FACRL(25),FACIM(25)
      PDRL=1.0
      PDIM=0.0
      DO 10 J=1,NMULT
      K=NC(J)
      CCRL=-FACRL(K)
      PDRL2=PDRL*CCRL-PDIM*FACIM(K)
      PDIM2=PDRL*FACIM(K)+PDIM*CCRL
      PDRL=PDRL2
   10 PDIM=PDIM2
      RETURN
      END
      SUBROUTINE PLYMPY (M,X,N,Y,MPN,Z)
C     PLYMPY MULTIPLIES TWO POLYNOMIALS, GIVING THE COEFFICIENTS OF
C     THE PRODUCT.
      DIMENSION X(25),Y(25),Z(25)
      MPN=M+N
      MPN1=MPN+1
      DO 70 K=1,MPN1
      L=K-1
      C=0.0
      DO 60 I=1,K
      J=I-1
      IF(M+J-L) 10,20,20
   10 A=0.0
      GO TO 60
   20 A=X(L-J)
   30 IF (N-J) 40,50,50
   40 B=0.0
      GO TO 60
   50 B=Y(J)
   60 C=C+A*B
   70 Z(L)=C
      RETURN
      END
      SUBROUTINE RHPCHK (KL,KND,KORD,ACRL,KRHP,KOUT)
C     RHPCHK CHECKS THE NUMERATOR AND DENOMINATOR OF THE OPEN LOOP
C     TRANSFER FUNCTION FOR ROOTS IN THE RIGHT HALF PLANE.
      DIMENSION ACRL(25)
      DO 10 I=1,KORD
      IF (ACRL(I)) 10,10,15
   10 CONTINUE
      IF (KOUT) 250,250,160
   15 KOUT=1
      IF (KL) 20,20,70
   20 IF (KND) 30,30,50
   30 PRINT 40
   40 FORMAT (1H0,//,6X,32H NOTE - PROGRAM HAS READ ZERO(S)    )
      GO TO 130
   50 PRINT 60
   60 FORMAT (1H0,//,6X,32H NOTE - PROGRAM HAS READ POLE(S)    )
      GO TO 130
   70 PRINT 80
   80 FORMAT (1H0,//,6X,32H NOTE - WITH COEFFICIENTS GIVEN,    )
      IF (KND) 90,90,110
   90 PRINT 100
  100 FORMAT (1H0,12X,25H PROGRAM COMPUTES ZERO(S)    )
      GO TO 130
  110 PRINT 120
  120 FORMAT (1H0,12X,25H PROGRAM COMPUTES POLE(S)    )
  130 PRINT 140
  140 FORMAT (1H0,12X,25H IN THE RIGHT HALF OF THE     )
      PRINT 150
  150 FORMAT (1H0,12X,15H COMPLEX PLANE.    )
      IF (KND) 250,250,160
  160 IF (KRHP) 170,170,250
  170 PRINT 180
  180 FORMAT(1H0,////,7X,42HERROR - PROGRAM WILL NOT PROCESS OPEN LOOP )
      PRINT 190
  190 FORMAT (1H0,13X,37H ZERO(S) OR POLE(S) IN THE RIGHT HALF    )
      PRINT 200
  200 FORMAT (1H0,13X,38H PLANE, UNLESS SPECIFICALLY TOLD TO DO    )
      PRINT 210
  210 FORMAT (1H0,13X,27H SO ON THE FIRST DATA CARD.    )
  250 RETURN
      END
      SUBROUTINE ROOTFND (KX,RAS,SARL,SAIM,IME)
C     ROOTFND FINDS THE ROOTS OF A POLYNOMIAL AND CHECKS THE ACCURACY
C     OF THE RESULT. USES SUBROUTINE RTPLSUB, NEWTON-RAPHSON AND
C     BAIRSTOW METHODS.
      DIMENSION RAS(25),SAR(25),SARL(25),SAIM(25),SARC(25)
      KXP=KX+1
      DO 5 I=1,KXP
      J=I-1
      IF (RAS(J)) 6,3,6
    3 KXJ=KX-J
      SARL(KXJ)=0.0
      SAIM(KXJ)=0.0
    5 CONTINUE
      GO TO 100
    6 KY=KX-J
      KYP=KY+1
      DO 10 I=1,KYP
      KZ=KXP-I
   10 SAR(I)=RAS(KZ)
      CALL RTPLSUB (KY,SAR,SARL,SAIM,IER)
      IF (IER) 15,15,100
   15 DO 20 I=1,KY
      SARMAG=ABSF(SARL(I))
      SAIMAG=ABSF(SAIM(I))*1.0E-06
      IF (SARMAG-SAIMAG) 17,17,20
   17 SARL(I)=0.0
   20 CONTINUE
      IF (IME) 100,25,25
   25 CALL DATAPR1 (KX,SARL,SAIM,1)
      MARK=0
      CALL RTARRG (KX,SARL,SAIM,MARK)
      CALL RTEXPD (KX,SARL,SAIM,SARC)
      KARM=0
      KP1=KY+1
      DO 40 I=1,KP1
      J=I-1
      SARDIF=0.01*ABSF (RAS(J))
      SARERR=ABSF (SARC(J)-RAS(J))
      IF (SARDIF-SARERR) 30,40,40
   30 KARM=1
   40 CONTINUE
      IF (IME) 50,50,60
   50 PRINT 52
```

```
 52 FORMAT (1H0,//,6X,43H NOTE - ACCURACY OF ROOT FINDING SUBROUTINE )
    PRINT 54
 54 FORMAT (1H0,12X,36H HAS BEEN CHECKED BY EXPANDING THESE    )
    PRINT 56
 56 FORMAT (1H0,12X,38H ROOTS, AND COMPARING THE COEFFICIENTS    )
    PRINT 58
 58 FORMAT (1H0,12X,38H OBTAINED WITH THE COEFFICIENTS OF THE    )
    PRINT 59
 59 FORMAT (1H0,12X,21H ORIGINAL POLYNOMIAL.    )
    GO TO 70
 60 PRINT 62
 62 FORMAT (1H0,//,7X,43HNOTE - ACCURACY HAS BEEN CHECKED, AS ABOVE. )
 70 IF (KARM) 80,80,90
 80 PRINT 82
 82 FORMAT (1H0,12X,37H ERROR NOTED IS LESS THAN ONE PERCENT   )
    PRINT 84
 84 FORMAT (1H0,12X,35H FOR ALL COEFFICIENTS. SOLUTION FOR    )
    PRINT 86
 86 FORMAT (1H0,12X,30H ROOTS IS CONSIDERED ACCURATE.    )
    GO TO 100
 90 PRINT 92
 92 FORMAT (1H0,12X,40H ERROR NOTED IS GREATER THAN ONE PERCENT    )
    PRINT 94
 94 FORMAT (1H0,12X,39H FOR AT LEAST ONE COEFFICIENT. SOLUTION    )
    PRINT 96
 96 FORMAT (1H0,12X,38H FOR ROOTS IS NOT CONSIDERED ACCURATE.    )
100 RETURN
    END
    SUBROUTINE RTPLSUB (N,A,U,V,IER)
    DIMENSION A(50),H(50),B(50),C(50),D(50),E(50),U(50),V(50),CONV(50)
    F=10.0
    E=25
    IER=0
    IF(N) 54,54,52
 54 IER=1
 52 NP3=N+3
```

```
100 B(2)=0.0
    B(1)=0.0
    C(2)=0.0
    C(1)=0.0
    D(2)=0.0
    E(2)=0.0
    H(2)=0.0
    DO 101 J=3,NP3
101 H(J)=A(J-2)
    T=1.0
    SK=10.0**F
150 IF(H(NP3)) 200,151,200
151 U(NP3)=0.0
    V(NP3)=0.0
    CONV(NP3)=SK
    NP3=NP3-1
    IF(NP3)152,152,150
152 IER=1
200 IF(NP3-3)205,51,201
205 IER=1
201 PS=0.0
    QS=0.0
    PT=0.0
    QT=0.0
    S=0.0
    REV=1.0
    SK=10.0**F
    IF(NP3-4)206,202,203
206 IER=1
202 R=-H(4)/H(3)
    GO TO 500
203 DO 207 J=3,NP3
    IF(H(J))204,207,204
204 S=S LOGF(ABSF(H(J)))
207 CONTINUE
    FPN1=N+1
```

```
      S=EXPF(S/FPN1)
      DO 208 J=3,NP3
  208 H(J)=H(J)/S
  210 IF(ABSF(H(4)/H(3))-ABSF(H(NP3-1)/H(NP3)))250,252,252
  250 T=-T
      M=(NP3-4)/2 + 3
      DO 251 J=3,M
      S=H(J)
      JJ=NP3-J+3
      H(J)=H(JJ)
  251 H(JJ)=S
  252 IF(QS) 253,254,253
  253 P=PS
      Q=QS
      GO TO 300
  254 HH2=H(NP3-2)
      IF(HH2) 256,255,256
  255 Q=1.0
      P=-2.0
      GO TO 257
  256 Q=H(NP3)/HH2
      P=(H(NP3-1)-Q*H(NP3-3))/HH2
  257 IF(NP3-5)258,550,258
  258 R=0.0
  300 DO 490 I=1,L
  350 DO 351 J=3,NP3
      B(J)=H(J)-P*B(J-1)-Q*B(J-2)
  351 C(J)=B(J)-P*C(J-1)-Q*C(J-2)
      IF(H(NP3-1))352,400,352
  352 IF(B(NP3-1))353,400,353
  353 AVHB1=ABSF(H(NP3-1)/B(NP3-1))
  356 IF(AVHB1-SK)450,354,354
  354 B(NP3)=H(NP3)-Q*B(NP3-2)
  400 IF(B(NP3))401,550,401
  401 AVHB2=ABSF(H(NP3)/B(NP3))
  403 IF(SK-AVHB2)550,450,450
```

```
450 DO 451 J=3,NP3
    D(J)=H(J)+R*D(J-1)
451 E(J)=D(J)+R*E(J-1)
    IF(D(NP3))452,500,452
452 AVHD3=ABSF(H(NP3)/D(NP3))
460 IF(SK-AVHD3)500,453,453
453 CC2=C(NP3-2)
    CC3=C(NP3-3)
    C(NP3-1)=-P*CC2-Q*CC3
    CC1=C(NP3-1)
    S=CC2*CC2-CC1*CC3
    IF(S)455,454,455
454 P=P-2.0
    Q=Q*(Q+1.0)
    GO TO 456
455 P=P+(B(NP3-1)*CC2-B(NP3)*CC3)/S
    Q=Q+(-B(NP3-1)*CC1+B(NP3)*CC2)/S
456 IF(E(NP3-1))458,457,458
457 R=R-1.0
    GO TO 490
458 R=R-D(NP3)/E(NP3-1)
490 CONTINUE
    PS=PT
    QS=QT
    PT=P
    QT=Q
    IF(REV)491,492,492
491 SK=SK/10.0
492 REV=-REV
    GO TO 250
500 IF(T)501,502,502
501 R=1.0/R
502 NP=NP3-3
    U(NP)=R
    V(NP)=0.0
    CONV(NP)=SK
```

```
      NP3=NP3-1
      DO 503 J=3,NP3
  503 H(J)=D(J)
      IF(NP3-3)300,51,300
  550 IF(T)551,552,552
  551 P=P/Q
      Q=1.0/Q
  552 PP2=P/2.0
      QMPSQ=Q-PP2*PP2
  560 IF(QMPSQ)554,554,553
  553 NP=NP3-3
      U(NP)=-PP2
      U(NP-1)=-PP2
      S=SQRTF(QMPSQ)
      V(NP)=S
      V(NP-1)=-S
      GO TO 561
  554 S=SQRTF(-QMPSQ)
      NP=NP3-3
      IF(P)555,556,556
  555 U(NP)=-PP2+S
      GO TO 557
  556 U(NP)=-PP2-S
  557 U(NP-1)=Q/U(NP)
      V(NP)=0.0
      V(NP-1)=0.0
  561 CONV(NP)=SK
      CONV(NP-1)=SK
      NP3=NP3-2
      DO 558 J=3,NP3
  558 H(J)=B(J)
      GO TO 200
   51 RETURN
      END
      SUBROUTINE SSERR1 (KLN,ERR,LFBK,TNHD,TDHD,TNHN,KTNO,KTDO,KTD1,
     1 ECDO,EKXNO,EKNO,MARK)
```

```
C     SSERR1 CHECKS THE VALIDITY OF THE STEADY STATE ERROR SPECIFICATION
C     AND COMPUTES THE COEFFICIENTS OF THE ERROR EQUATION.
      DIMENSION TNHD(11),TDHD(11),TNHN(11)
      IF (KTDO) 20,20,60
   20 IF (LFBK) 120,120,30
   30 PRINT 40
   40 FORMAT (1H0,//,6X,42H ERROR - WITH POLE(S) AT THE ORIGIN OF THE  )
      PRINT 42
   42 FORMAT (1H0,13X,33H OPEN LOOP TRANSFER FUNCTION, AND   )
      PRINT 44, LFBK
   44 FORMAT (1H0,13X,2H ( ,I1,27H) ORDER FEEDBACK SPECIFIED,  )
      PRINT 50
   50 FORMAT (1H0,13X,32H STEADY STATE ERROR IS INFINITY.  )
      PRINT 55
   55 FORMAT (1H0,13X,45H STEADY STATE ERROR SPECIFICATION IS INVALID. )
      MARK=1
      GO TO 130
   60 IF (KTNO) 70,70,120
   70 PRINT 80
   80 FORMAT(1H0,//,7X,45HNOTE - WITH ZERO(S) AT THE ORIGIN OF THE OPEN)
      PRINT 82
   82 FORMAT (1H0,12X,37H LOOP TRANSFER FUNCTION, STEADY STATE   )
      PRINT 84
   84 FORMAT (1H0,12X,32H ERROR IS EQUAL TO ONE.  (ESS=1)   )
      EMEAS=ERR*1.0E-02
      SPECE=ABSF(1.0-ERR)
      IF (SPECE-EMEAS) 110,110,90
   90 PRINT 100
  100 FORMAT (1H0,//,6X,41H ERROR - STEADY STATE ERROR SPECIFICATION   )
      PRINT 102
  102 FORMAT (1H0,13X,13H IS INVALID.   )
      MARK=1
      GO TO 130
  110 KLN=0
      EKNO=1.0
      EKXNO=-1.0
```

```
114 ECDO=0.0
    PRINT 115
115 FORMAT (1H0,12X,34H STEADY STATE ERROR IS INDEPENDENT    )
    PRINT 116
116 FORMAT (1H0,12X,27H OF THE SYSTEM PARAMETERS.  )
    GO TO 125
120 ECDO=(ERR-1.0)*TDHD(0)
    EKXNO=(ERR-1.0)*TNHN(0)
    EKN =TNHD(0)
125 MARK=0
130 RETURN
    END
    SUBROUTINE BANDSLV (N1N,TNHD,N1D,TDHD,N2D,TNHN,ECD0,EKXN0,EKN0,
   1 JRSPC,FCK,FCJ,BASE,FAQ,RTRL,RTIM,BOMEG,GMGN,GMGD,GMGT,MARK)
C     BANDSLV SOLVES THE SYSTEM FOR THE STEADY STATE ERROR
C     SPECIFICATION, AND COMPUTES THE BANDWIDTH.
    DIMENSION TNHD(11),TNHN(11),TDHD(11),RTRL(25),RTIM(25),BAR(25),
   1 COLE(25,25),VAR(25),CN(10),CD(10),GN(25)
    IMX=(2*N1D)-1
 10 CALL ROOTGEN (JRSPC,IMX,FCK,FCJ,BASE,FAQ,RTRL,RTIM)
 20 CALL RTEXPD (IMX,RTRL,RTIM,BAR)
    CALL BARARRG (IMX,BAR)
    CALL COPLUG (N1D,TDHD,N2D,TNHN,NMAT,COLE)
    LX1=NMAT+1
    LX2=N1D+1
    COLE(LX1,1)=ECD0
    COLE(LX1,LX2)=EKXN0
    COLE(LX1,LX1)=EKN0
    CALL LINSOLV (LX1,COLE,BAR,VAR,MARK)
    IF (MARK) 30,30,40
 30 CALL PARSOLV (N1D,VAR,CN,CD,GKX,GKT)
    CALL RTEXPD (IMX,RTRL,RTIM,BAR)
    CALL BANDSL1 (N1N,TNHD,N1D,CN,GKT,BAR,BOMEG,GMGN,GMGD,GMGT)
 40 RETURN
    END
    SUBROUTINE ROOTGEN (IMIN,IMAX,FCK,FCJ,BASE,FAQ,FACRL,FACIM)
```

```
C     ROOTGEN GENERATES THE ADDITIONAL ROOTS REQUIRED FOR SOLUTION
C     0F THE SYSTEM OF SIMULTANEOUS LINEAR EQUATIONS.
      DIMENSION FACRL(25),FACIM(25)
      IRGEN=IMAX-IMIN
      AM=O.0
      IF (IRGEN-1) 40,35,5
    5 IF (IRGEN-3) 10,15,15
   10 IXM1=IMAX-1
      FACRL(IXM1)= FCK*BASE
      FACIM(IXM1)= 0.0
      GO TO 35
   15 IRG1=IRGEN/2
      IRG2=2*IRG1
      INP=IMIN+1
      MFAC=INP
      IF (IRGEN-IRG2) 20,20,25
   20 FACRL(INP)= FCK*BASE
      FACIM(INP)=0.0
      INP=INP+1
   25 IXM=IMAX-2
      DO 30 I=INP,IXM,2
      IP1=I+1
      IFC=I-MFAC
      IFC=(IFC+1)/2
      AM=IFC
      AMF=((AM*FCJ)+FCK)*BASE
      FACRL(I)=AMF
      FACRL(IP1)=AMF
      FACIM(I)=FAQ
   30 FACIM(IP1)=-FAQ
   35 FACRL(IMAX)=(((AM+1.0)*FCJ)+FCK)*BASE
      FACIM(IMAX)= 0.0
   40 RETURN
      END
C     SUBROUTINES BARARRG AND REVARRG ARRANGE THE COEFFICIENTS OF
C     THE CHARACTERISTIC EQUATION FOR USE IN OTHER SUBROUTINES.
```

```
      SUBROUTINE BARARRG (MAX,BAR)
      DIMENSION BAR(25)
      MAXP1=MAX+1
      MAXP2=MAX+2
      DO 10 I=1,MAXP1
      J=MAXP2-I
      K=J-1
      BAR(J)=BAR(K)
   10 BAR(K)=0.0
      BAR(MAXP2)=0.0
      RETURN
      END
      SUBROUTINE REVARRG (MAX,BAR)
      DIMENSION BAR(25)
      MAXP1=MAX+1
      MAXP2=MAX+2
      DO 10 I=1,MAXP1
      J=I-1
      BAR(J)=BAR(I)
   10 BAR(I)=0.0
      BAR(MAXP2)=0.0
      RETURN
      END
      SUBROUTINE COPLUG (MHD,HAD,MHN,HAN,NMAT,COLE)
C     COPLUG FORMS THE COEFFICIENT MATRIX FOR THE SOLUTION OF THE
C     SYSTEM OF LINEAR EQUATIONS.
      DIMENSION HAD(11),HAN(11),COLE(25,25)
      MHNP1=MHN+1
      MHDP1=MHD+1
   10 DO 15 I=1,25
      DO 15 J=1,25
   15 COLE(I,J)=0.0
      DO 20 I=MHNP1,11
   20 HAN(I)=0.0
      DO 25 I=MHDP1,11
   25 HAD(I)=0.0
```

```
      NMAT=MHD*2
      DO 45 I=1,NMAT
      LSTOP=I
      IF (I-MHD) 35,35,30
   30 LSTOP=MHD
   35 DO 45 J=1,LSTOP
      JPM=J+MHD
      IMJ=I-J
      IF (IMJ-MHD) 40,40,45
   40 COLE(I,J)=HAD(IMJ)
      COLE (I,JPM)=HAN(IMJ)
   45 CONTINUE
   50 RETURN
      END
      SUBROUTINE LINSOLV (NEQ,COLE,BAR,VAR,MARK)
C     LINSOLV SOLVES THE SYSTEM OF SIMULTANEOUS LINEAR EQUATIONS
C     BY THE GAUSS ELIMINATION METHOD.
      DIMENSION COLE(25,25),BAR(25),ROLE(25),VAR(25)
      NEQM1=NEQ-1
      DO 50 K=1,NEQM1
      KP1=K+1
      NMARK=NEQ-K
      NCOUNT=0
      IF (COLE(K,K)) 30,10,30
   10 DO 27 I=KP1,NEQ
      IF (COLE(I,K)) 15,25,15
   15 DO 20 J=K,NEQ
      HOLDCOL=COLE(K,J)
      COLE(K,J)=COLE(I,J)
   20 COLE(I,J)=HOLDCOL
      HOLDBAR=BAR(K)
      BAR(K)=BAR(I)
      BAR(I)=HOLDBAR
      GO TO 30
   25 NCOUNT=NCOUNT+1
   27 CONTINUE
```

```
      IF (NCOUNT-NMARK) 30,55,55
   30 ABLE=COLE(K,K)
      DO 35 J=K,NEQ
      COLE(K,J)=COLE(K,J)/ABLE
   35 ROLE(J)=COLE(K,J)
      BAR(K)=BAR(K)/ABLE
      BOLE=BAR(K)
      DO 50 I=KP1,NEQ
      IF (COLE(I,K)) 40,50,40
   40 ABLE=COLE(I,K)
      DO 45 J=K,NEQ
      COLE(I,J)=COLE(I,J)/ABLE
      COHOLD=COLE(I,J)
      COLE(I,J)=ROLE(J)-COHOLD
      CMIN=ABSF(ROLE(J))*1.0E-07
      CIM=ABSF(COLE(I,J))
      IF (CIM-CMIN) 42,42,44
   42 COLE(I,J)=0.0
   44 ROLE(J)=COHOLD
   45 CONTINUE
      BAR(I)=BAR(I)/ABLE
      BARHOLD=BAR(I)
      BAR(I)=BOLE-BARHOLD
   47 BOLE=BARHOLD
   50 CONTINUE
      IF (COLE(NEQM1,NEQM1)) 65,55,65
   55 PRINT 60
   60 FORMAT (1H0,//,15X,35H SOLUTION OF THE COEFFICIENT MATRIX  )
      PRINT 61
   61 FORMAT (1H0,//,6X,43H NOTE - THE COEFFICIENT MATRIX IS SINGULAR, )
      PRINT 62
   62 FORMAT (1H0,12X,40H PROGRAM WILL NOT SOLVE FOR THIS SYSTEM.  )
      MARK=1
      GO TO 80
   65 VAR(NEQ)=BAR(NEQ)/COLE(NEQ,NEQ)
      MARK=0
      DO 75 I=1,NEQM1
      J=NEQ-I
      JP1=J+1
      A=0.0
      DO 70 K=JP1,NEQ
   70 A=A COLE(J,K)*VAR(K)
   75 VAR(J)=BAR(J)-A
   80 RETURN
      END
      SUBROUTINE PARSOLV (N1D,VAR,CN,CD,GKX,GKT)
C     PARSOLV USES THE GROUPED VARIABLES FOUND BY SUBROUTINE LINSOLV
C     TO SOLVE FOR VALUES OF THE INDIVIDUAL PARAMETERS.
      DIMENSION VAR(25),CN(10),CD(10)
      DO 10 I=1,N1D
      J=I-1
   10 CD(J)=VAR(I)
      N5D=2*N1D
      N1DP1=N1D+1
      CON=VAR(N5D)
      DO 20 I=N1DP1,N5D
      J=I-N1DP1
   20 CN(J)=VAR(I)/CON
      N5DP1=N5D+1
      GKX=VAR(N1DP1)/VAR(N5DP1)
      GKT=VAR(N5D)/GKX
      RETURN
      END
C     SUBROUTINES BANDSL(1 AND 3) COMPUTE THE BANDWIDTH OF THE
C     CLOSED LOOP SYSTEM.
      SUBROUTINE BANDSL1 (N1N,TNHD,NCORD,CN,GKT,BAR,BOMEG,GMGN,GMGD,
     1 GMGT)
      DIMENSION TNHD(11),CN(10),BAR(25),GN(25)
      NCM=NCORD-1
      CALL PLYMPY (N1N,TNHD,NCM,CN,N4D,GN)
      CALL BANDSL3 (N4D,GN,BOMEG,GMGN)
      IMX= (2*NCORD) -1
   CALL BANDSL3 (IMX,BAR,BOMEG,GMGD)
   GMGT=(GKT*GMGN)/GMGD
   RETURN
   END
   SUBROUTINE BANDSL3 (MAX,CLN,BOMEG,GMAG)
   DIMENSION CLN(25)
   IF (MAX-1) 2,4,5
 2 GMAG=CLN(0)
   GO TO 35
 4 GRN=CLN(0)
   GIN=BOMEG*CLN(1)
   GO TO 33
 5 MAX1=MAX/2
   MAX2=2*MAX1
   IF (MAX-MAX2) 10,10,15
10 MAXE=MAX
   MAXO=MAX-1
   GO TO 20
15 MAXO=MAX
   MAXE=MAX-1
20 GRN=CLN(0)
   DO 25 I=2,MAXE,2
   J=I/2
   ISN=(-1)**J
   AI=ISN
25 GRN=GRN+(CLN(I)*AI*(BOMEG**I))
   GIN= CLN(1)*BOMEG
   DO 30 I=3,MAXO,2
   J=(I-1)/2
   ISN=(-1)**J
   AI=ISN
30 GIN=GIN+(CLN(I)*AI*(BOMEG**I))
33 GMAG=SQRTF ((GRN**2)+(GIN**2))
35 RETURN
   END
   SUBROUTINE ITERPR (NCT,MCT,GMDB,BDSPC,RLER)
```

```
C       ITERPR IS A PRINTOUT FORMAT CHECKING CONVERGENCE FOR THE
C       BANDWIDTH SOLUTION.
        IF (NCT) 10,10,35
     10 PRINT 15
     15 FORMAT (1H0,//,11X,42HITERATIVE PROCEDURE FOR BANDWIDTH SOLUTION )
        IF (NCT) 21,17,17
     17 PRINT 20
     20 FORMAT (1H0,/,10X,22H PROCESS IS CONVERGING    )
     21 IF (BDSPC) 27,22,27
     22 PRINT 25
     25 FORMAT (1H0,/,10X,22H ITERATION     GAIN-DB    )
        GO TO 37
     27 PRINT 30
     30 FORMAT (1H0,/,10X,40H ITERATION     GAIN-DB    RELATIVE ERROR    )
     35 IF (BDSPC) 40,37,40
     37 PRINT 38, MCT,GMDB
     38 FORMAT (1H0,/,10X,I7,7X,1PE9.2)
        GO TO 55
     40 GMPC=GMDB-BDSPC
        RLER=(100.0*GMPC)/ABSF(BDSPC)
        PRINT 50, MCT,GMDB,RLER
     50 FORMAT (1H0,/,10X,I7,7X,1PE9.2,3X,1PE9.2,8H PERCENT    )
     55 RETURN
        END
        END
```

1934

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2.	Library Naval Postgraduate School Monterey, California 93940	2
3.	Commander, Naval Ship Systems Command Department of the Navy Washington, D. C.20360	1
4.	Professor G. J. Thaler Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	5
5.	LT Michael J. Breslin, USN USS MARS (AFS-1) FPO San Francisco, California 96601	2
6.	LT Albert B. Lemanski, USN USS NORTON SOUND (AVM-1) FPO San Francisco, California 96601	1
7.	Professor A. Gerba Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
8.	Professor D. D. Siljak Department of Electrical Engineering University of Santa Clara Santa Clara, California 95053	1
9.	Professor Toyomi Ohta Electrical Department, Defense Academy Obaradai, Yokosuka, Japan	1
10.	LT C. H. Hyon Division of Control Engineering University of Saskatchewan Saskatoon, Saskatchewan, Canada	1
11.	Dr. K. W. Han Institute of Electronics National Chiao-Tung University Hsinchu, Taiwan	1

12. Dr. S. H. Liu 1
P. O. Box 1
Shih-yi-Fen
Lungtan, Taiwan

13. Mr. Seltzer 1
Code R - ASTR - NG
NASA George Marshall Space Flight Center
Huntsville, Alabama 35812

14. Mr. D. R. Towill 1
Welsh College of Advanced Technology
Cathays Park, Cardiff, Wales

15. Dr. A. G. Thompson 1
Department of Mechanical Engineering
University of Adelaide
Adelaide, Australia

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

COMPUTER TECHNIQUES FOR SYSTEM SYNTHESIS

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

M.S. Thesis - Electrical Engineering - 1967

5. AUTHOR(S) *(Last name, first name, initial)*

BRESLIN, Michael J.
Lieutenant, United States Navy

6. REPORT DATE June 1967	7a. TOTAL NO. OF PAGES 130	7b. NO. OF REFS 2

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. AVAILABILITY/LIMITATION NOTICES

[redacted]

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Naval Ship Systems Command Navy Department Washington, D. C. 20360

13. ABSTRACT

A major hindrance to the use of algebraic techniques for the design of linear control systems is the non-linearity of the equations describing the system. The equations for a general class of control systems are shown to be linear in certain groupings of variables contained in, or added to, the systems. A design technique based on the solution of these linear equations is outlined, and a digital computer program which will accomplish the design is presented.

DD FORM 1 JAN 64 1473

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Non-linear systems						
Algebraic techniques						
Grouped variables						
Individual parameters						

DD FORM 1 NOV 65 1473 (BACK)
S/N 0101-807-6821

A-31409

thesB8038
Computer techniques for syste[illegible]thesis
DUDLEY KNOX LIBRARY
3 2768 00417083 7
DUDLEY KNOX LIBRARY